

## مقدمه

اینترنت به عنوان ابزاری که بازیابی سریع و آسان اطلاعات را فراهم می آورد از سوی کاربران بسیاری مورد استفاده و استناد قرار می گیرد. کاربران معمولاً بدون توجه به معیارهای ارزیابی انتقادی هر آنچه را که در اینترنت بازیابی می شود معتبر می شمارند و در مطالعات تحقیقاتی خویش مورد استفاده و استناد قرار می دهند. (Case, ۲۰۰۳) نظر قالب کتابداران و اعضای هیات علمی آن است که استناد در تحقیقات دانشجویی به سمت منابع اینترنتی رفته و دانشجویان بر استفاده از این منابع بسیار تاکید دارند. (Davis, ۲۰۰۲, نقل از 4: Tillotson, 2003) یافته های تولپانن نشان می دهد که ۹۲/۵ درصد از دانشجویان ورودی زبان انگلیسی اطلاعات موجود در اینترنت را معتبر می دانند و این در حالی است که ۵۲/۵ درصد از آنان اطلاعات بازیابی شده را هرگز ارزیابی نمی کنند. (Tillotson, 2003: 4 نقل از Tolpannan, 1999)

اغلب کاربران اطلاعات بازیابی شده از اینترنت را معتبر می شمارند و قادر به تشخیص گهر از زباله نیستند. اغلب کاربران در پاسخ به این سوال که چرا به این منبع استناد کرده اید و آیا این منبع معتبر است یا خیر؟ چنین پاسخ می دهند که "آن را جستجو و بازیابی کرده ام و از آن استفاده می کنم." در این پاسخ جایی برای ارزیابی منتقدانه اطلاعات بازیابی شده گذاشته نشده است و اتکای صرف به مطالب بازیابی شده و ارزیابی نشده به طور وحشتناکی نمایان است. (Williams, ۲۰۰۳) تسهیلات تکنولوژیکی موجود بسیاری از موانع انتشار و اشاعه اطلاعات را از میان برداشته است. امروز هر فردی می تواند به نشر عقاید خویش پرداخته و وب سایت خود را مدیریت نماید. گستره صفحات وب روز به روز بیشتر می شود و بر آلودگی اطلاعات در این آشفته بازار اطلاعاتی افزوده می گردد. از سویی جامعه امروز به سمت جامعه اطلاعاتی در حال حرکت است. جهانی شدن و اطلاعات گرایی از مشخصه های عصر اطلاعات است. دسترسی به اطلاعات متناسب با نیازهای اطلاعاتی، در زمانی کوتاه مهارتهایی خاص می طلبد. سواد اطلاعاتی لازمه جامعه اطلاعاتی است و برای رسیدن به این سواد کاربران باید به مهارتهای اطلاع یابی مجهز باشند. یکی از مهارتهای اطلاع یابی، مهارت ارزیابی است. اما پرسش آن است که آیا واقعا اطلاعات بازیابی شده از اینترنت نیاز به ارزیابی دارند و یا هر نوع اطلاعات بازیابی شده می تواند مستند باشد و مورد استناد قرار گیرد؟ ضرورت ارزیابی اطلاعات بازیابی شده در چیست و چه معیارهایی برای ارزیابی وجود دارد؟ در این نوشته سعی بر آن است تا با روشن ساختن ضرورت ارزیابی اطلاعات بازیابی شده در فضای اینترنت، معیارهای ارزیابی صفحات وب به عنوان یکی از محمل های اطلاعاتی تبیین گردد.

جهت خرید فایل word به سایت [www.kandooocn.com](http://www.kandooocn.com) مراجعه کنید  
یا با شماره های ۰۹۳۶۶۰۲۷۴۱۷ و ۰۹۳۶۶۴۰۶۸۵۷ و ۰۶۶۴۱۲۶۰-۵۱۱ تماس حاصل نمایید

فصل اول:

بیان ضرورت و معیارهای ارزیابی در صفحات وب

## ۱-۱ مهارت ارزیابی اطلاعات

کتابداران و اطلاع رسانی از دیرباز در پی افزایش ارتباط سودمند میان منابع اطلاعاتی و طالبان آنها بوده اند و در جهت افزایش این ارتباط موثر و افزایش خرد و دانایی در سطح جامعه همواره تلاش کرده اند. اطلاعات صرفنظر از محملی که بر بستر آن قرار خواهد گرفت دارای ارزش فراوان است و به خلق دانایی می انجامد. اطلاع جویی فرایندی هدفمند و ساختارمند است. طرح نیاز اطلاعاتی، تعریف استرژیک جستجو، بازیابی اطلاعات، ارزیابی آنچه بازیابی شده است، سازماندهی و استفاده موثر از اطلاعات در جهت تبادل اندیشه ها و یافته ها، مهارتهایی می طلبد. این مهارتها در قالب واژه مهارتهای اطلاع یابی جای می گیرند.

## ۱-۲ مفهوم مهارت اطلاع یابی

یان مال لی (لی، یان مال، ۱۳۷۶) مهارت اطلاع یابی را اینگونه بیان میدارد:

۱. مهارت در بازیابی اطلاعات

۲. مهارت در ارزیابی اطلاعات

۳. مهارت در سازماندهی اطلاعات و

۴. مهارت در تبادل اطلاعات

ارزیابی اطلاعات به عنوان یکی از کلیدی ترین مهارتهای مورد نیاز کاربران در این تقسیم بندی مورد اشاره است. این مهارت در انتخاب منابع متناسب با نیازهای اطلاعاتی کارساز است و روند اطلاع یابی موثر را سرعت و اعتبار می بخشد. اطلاعات پس از بازیابی در هر قالب و ساختاری که ارائه شود نیاز به ارزیابی دارد تا به این وسیله محتوای اطلاعاتی، متناسب با نیازهای تعریف شده کاربران باشد و به حل مسئله بیانجامد. کاربران جهت ارزیابی اطلاعات بازیابی شده نیازمند تفکر انتقادی هستند.

## ۱-۳ مفهوم تفکر انتقادی

خمیر مایه تفکر انتقادی، تکاپوی ذهنی و جستجو در اندیشه های نوشته و نانوشته جهت دستیابی به نتیجه ای معتبر و مفید است. این گونه تفکر آمیزه ای از تفکر خلاق، تفکر منطقی و تفکر مسئله گشاست. تفکر مسئله گشا همه عوامل ذیربط را در تفکر انتقادی وارد میسازد، تفکر منطقی، راه نفوذ خطاهای منطقی را در تفکر انتقادی سد می کند و تفکر خلاق ابزار لازم را برای ترکیب مفاهیم به گونه ای متفاوت از گذشته فراهم می آورد. (دیانی، ۱۳۷۰).

تفکر انتقادی لازمه توانایی و مهارت ارزیابی اطلاعات است. نظام آموزشی و کتابخانه ها به عنوان زیرمجموعه ای از این نظام، وظیفه تربیت کاربرانی مجهز به تفکر انتقادی را بر عهده دارند. حری معتقد است که نظام آموزش عالی دو نقش دارد یکی اینکه به دانشجویان پیاموزد چگونه اطلاعات خویش را

بیابد و دیگری اینکه به آنان تعلیم دهد چگونه اطلاعات به دست آمده را تعبیر و تفسیر کند. (حری، ۱۳۷۸)

تفکر انتقادی در محیط آموزشی از جمله کتابخانه ها شکل می گیرد و پرورش می یابد، لذا لازم است کتابخانه ها در ارائه خدمات خود به ایجاد و تقویت نگاه انتقادی در کاربران توجه داشته باشند و در آموزشهای کتابخانه ای خود این جنبه را لحاظ کنند. وجود وب سایتهای کتابخانه ای که آموزش مهارتهای اطلاع یابی را به صورت مجازی فراهم می آورد فرصت مناسبی برای این منظور است. اما هنوز سوالی باقیست: آیا ارزیابی اطلاعات ضروری است؟

#### ۱-۴- ضرورت ارزیابی اطلاعات در عصر اطلاعات:

آدمی برای رفع نیاز اطلاعاتی خویش به دنبال کسب اطلاعات است. وی از اطلاعاتی که دریافت میکند برای تصمیم گیریها و ارتقای دانش خویش بهره می برد. تصمیم گیری زمانی ارزشمند است که به درستی صورت پذیرد. نباید چنین پنداشت که تمامی اطلاعات کسب شده از صحت و اعتبار لازم برای دخیل شدن در تصمیم گیریها برخوردارند. در اینجا است که اهمیت ارزیابی اطلاعات رخ می نماید. از طرفی جامعه اطلاعاتی در عصر اطلاعات ضرورتهای دیگری را نیز تحمیل می کند:

۱-۴-۱ جهانی شدن و ارزش اطلاعات: یکی از نتایج بحث جهانی شدن، افزایش ارزش اطلاعات به عنوان معیار توسعه یافتگی و خمیر مایه توسعه پایدار ملتهاست لذا روز به روز بر ارزش اطلاعات افزوده می شود.

۱-۴-۲ رشد تکنولوژیکی و آلودگی اطلاعات: رشد سریع شبکه های اطلاعاتی روند اشاعه اطلاعات را تسهیل نموده، متخصصان موضوعی راحت تر از گذشته به تبادل اطلاعات می پردازند و چرخه تولید علم سریع تر از گذشته در جریان است. در نتیجه آلودگی اطلاعات روز به روز افزایش و طول عمر مفید اطلاعات کاهش می یابد.

۱-۴-۳ اطلاع گرایی و جامعه اطلاعاتی: مفهوم جامعه اطلاعاتی در حال شکل گیری است. در جامعه اطلاعاتی کلیه تصمیم گیری ها بر مبنای اطلاعات صورت می گیرد و سرعت دسترسی به اطلاعات روزآمد، صحیح و معتبر، ارزشمند است. کاربران از میان انبوه اطلاعات تولید شده بر مبنای نیاز اطلاعاتی خویش به انتخاب دست می زنند و به تصمیم گیری های مفید و به موقع می پردازند.

۱-۴-۴ کاربر نهایی و مهارتهای اطلاع یابی: مفهوم ارتباطات بی سیم کاربران را از محدودیت های زمانی و مکانی خارج ساخته و رفتار اطلاع یابی را به حریم خصوصی آنها کشانده است. مفهوم کاربر نهایی شکل گرفته و کاربران بدون یاری کتابداران به رفع نیازهای اطلاعاتی خویش می پردازند. در این صورت نیاز به آموزشهای پیچیده تری جهت ایجاد و ارتقای سطح مهارتهای اطلاع یابی احساس میشود.



## ۱-۵ ضرورت ارزیابی صفحات وب

از دیرباز تا امروز که در عصر اطلاعات به سر می بریم همواره ارزیابی اطلاعات و تقویت تفکر انتقادی مطرح بوده است. عصر اطلاعات نه تنها از ضرورت آن نکاسته که بر ضرورت آن افزوده است. محتوای اطلاعاتی ممکن است در هر قالبی قرار گیرد. در انواع رسانه ها صرفنظر از نوع محمل، محتوای اطلاعات مورد ارزیابی قرار میگیرد. صفحه وب سندی چندرسانه ایست که در محمل جدید اطلاعاتی یعنی وب قرار گرفته است. موتورهای جستجو با نمایه سازی صفحات وب، ارزیابی اطلاعات را در این محیط میسر می سازند. اما محیط مجازی وب ویژگی هایی خاص دارد که آن را از سایر رسانه ها متمایز می کند. به بیانی محیط مجازی وب بر محتوای اطلاعاتی تاثیر می گذارند.

بسیاری از متخصصان اطلاع رسانی و کتابداران بر این عقیده اند که ارائه اطلاعات در صفحات وب از اعتبار آنها خواهد کاست و هنوز فرمت چاپی را ترجیح میدهند این گروه بر این عقیده اند که اطلاعات در شکل چاپی از قابلیت اعتماد بیشتری برخوردار است چرا که مقالات در نشریات علمی پس از بررسی محتوایی چاپ میشوند.

بسیاری از اطلاعات با ارزش علمی در شبکه قرار نمی گیرند و کتابخانه ها مجلات علمی معتبری دارند که محتوای اطلاعاتی آنها به صورت تمام متن در اینترنت وجود ندارد. (Standler, 2004) این در حالی است که ناشران مجلات علمی به فکر تغییر بستر اطلاعاتی خویش افتاده و در اندیشه فراهم آوری امکان دسترسی تمام متن به مقالات علمی خویش هستند. مزایایی چون قابلیت دسترسی پذیری فراوان بدون محدودیت زمانی و مکانی، امکان روز آمد کردن سریع اطلاعات و قابلیت انتقال و کپی برداری سریع و به صرفه از اطلاعات در محیط اینترنت عواملی است که به روند دیجیتالی کردن اطلاعات سرعت می بخشد. در این صورت ارائه خدمات به صورت پولی انجام خواهد شد و نمونه این گونه از مجلات علمی در حال حاضر به صورت الکترونیکی در شبکه مشاهده می شود. (Standler, 2004) نوع محمل اطلاعاتی، در مورد صفحات وب بر محتوای اطلاعاتی آن تاثیر می گذارد لذا به صورت اختصاصی به بیان ضرورت ارزیابی اینگونه منابع پرداخته می شود:

- صفحات وب به سادگی ایجاد میشوند.
- دریافت دامنه و URL آسان است. حتی دامنه هایی که به ظاهر نشانگر مشخصه های خاصی چون org, edu, ... هستند با پرداخت مبلغ ناچیزی واگذار می شوند.
- هر صفحه ایجاد شده بدون هیچ محدودیتی از سوی موتورهای جستجو نمایه و آماده ارزیابی می شود.
- این صفحات می توانند یکی از منابع اطلاعاتی محسوب شده و مبنای استناد قرار گیرند.

• نظارتی بر محتوای سایتها و نوع اطلاعات ارائه شده در آنها وجود ندارد. و به بیانی هر گونه اطلاعاتی از طریق صفحات وب ارائه میشود.

• اطلاعات در این صفحات بدون هیچ گونه فیلتری و بدون توجه به ویژگی های کاربران در دسترس گروههای مختلف قرار می گیرد.

• داوری آنگونه که در مورد منابع چاپی مطرح است، در این دست از منابع به چشم نمی خورد.

• دنیای مجازی ارتباطات مبتنی بر وب، اعتماد پذیری را کاهش می دهد. چراکه هویت طراحان صفحات وب می تواند پنهان باشد و یا اطلاعات ارائه شده از سوی آنها کاملاً ساختگی باشد.

• اطلاعات ارائه شده در صفحات وب ناپایدار است.

• اطلاعات ارائه شده در وب از نظر محتوایی گاه بیهوده و بالقوه خطرناک و گمراه کننده است.

• به علت آلودگی بیش از حد اطلاعات در وب، میزان جامعیت بازیابی ها بسیار زیاد است نتیجه این آلودگی وجود سرریز اطلاعات است و برای رسیدن به نتیجه مطلوب باید به گزینش دست زد.

• به علاوه روشهای نمایه سازی در وب از کارایی لازم برخوردار نیستند و همواره بخش مفیدی از وب به صورت وب پنهان باقی می ماند.

شاید بتوان از طریق اعمال جستجوی پیشرفته بر غنای محتوایی اطلاعات بازیابی شده افزود اما در هر صورت در نمایه سازی وب، روابط معنایی آنگونه که شایسته است مد نظر نیست و چه بسا موارد بازیابی شده با اطلاعات مورد نظر سازگاری نداشته باشد. حتی با این فرض که موتور جستجو نمایه سازی را به درستی انجام داده باشد، طراحان وب در طراحی خویش ابر داده ها را مد نظر داشته باشند و موارد بازیابی شده مرتبط نیز باشند، لازم است که کاربران مطالب بازیابی شده را با دیدی انتقادی مورد ارزیابی قرار دهند و به این ترتیب بر اعتبار استناد خویش بیافزایند.

لذا کاربران چه به عنوان اعضای جامعه اطلاعاتی در عصر اطلاعات و چه به عنوان مراجعان ساده کتابخانه های قرون وسطی هیچ گاه حق ارزیابی منتقدانه را از خود سلب نکرده اند. ساختار ارائه اطلاعات چه چاپی باشد، چه میکرو فیلم و چه ابزارهای دیجیتالی، کاربران همواره معیارهایی را در ارزیابی های خویش به کار می برند.

در مورد منابع الکترونیکی و آنچه در شبکه های پیوسته اشاعه می یابد ضرورت ارزیابی نه تنها کاهش نمی یابد که پررنگ تر خواهد شد و تابع معیارهای ویژه ای است.

## ۱-۶ ارزیابی صفحات وب

برای یافتن اطلاعات مناسب با نیاز اطلاعاتی از طریق صفحات وب باید در دو چیز مهارت داشت:

• دستها و چشمها را به تکنیک هایی مجهز کنید که قادر به یافتن مناسب ترین اطلاعات در زمانی کوتاه

باشند.

● ذهن خود را طوری تربیت کنید که با دیدی انتقادی و حتی بدبینانه مجموعه ای از سوالات را طراحی و در مواجه با اطلاعات بازیابی شده آنها را در ارزیابی به کار بندد.  
در صورتیکه این دو فرایند همزمان انجام پذیرد می توان گفت که کاربر از مهارت بازیابی و ارزشیابی منابع مرتبط با نیاز خود برخوردار است.

## ۱-۷ معیارهای ارزیابی صفحات وب

برخی معیارهای ارزیابی صرفنظر از نوع محمل اطلاعاتی برای انواع منابع چاپی و غیر چاپی صادق اند اما برخی از معیارها اختصاص به نوعی از منابع دارند. اندیشه بنیانی در ارزیابی اطلاعات همچنان ثابت است اما گاهی اتفاق می افتد که با توجه به نوع کاربران، نوع و عمق نیاز اطلاعاتی و موارد استفاده از اطلاعات بازیابی شده برخی از معیارها پررنگ تر و برخی کم رنگ تر است.

محتوای این معیارها حول چند نکته می چرخد. ارزیابی از دو جهت کمی و کیفی باید انجام شود. نوع و ساختار سازماندهی و ارائه اطلاعات در انواع محمل ها به سهولت کاربری و استفاده موثر می افزاید لذا نباید چنین پنداشت که ساختار و شکل ارائه اطلاعات در کم اهمیت است. در مورد صفحات وب نیز هر دو جنبه محتوای کمی و کیفی مد نظر است.

## ۱-۸ هدف را مشخص کنید

آنچه در اینترنت منتشر می شود با هدف و برای مخاطبان خاصی است. شناسایی هدف با دنبال کردن پیوندهایی چون "Philosophy", "About us", "Who am I", "Biography", "Background". آشکار می شود. این پیوندها اهداف، دامنه، نوع فعالیت، وابستگی سازمانی و اطلاعات سودمند دیگری را مشخص می نماید. اگر در صفحه بازیابی شده این پیوندها را پیدا نکردید باید با کوتاه کردن URL به صفحه خانگی سایت رفته و در آنجا به دنبال اطلاعاتی از این دست باشید. (Barker, 2004). همواره به خاطر داشته باشید که هدف زیربنای سایر برنامه ریزی ها و طراحی هاست و در تعریف ملاکها و معیارهای گردآوری، تنظیم و ارائه اطلاعات تاثیر گذار است.

راه دیگر برای کشف هدف توجه به دامنه و URL صفحه بازیابی شده است. صفحاتی که با دامنه های mil, gov, org, edu, مشخص می شوند از اعتبار بیشتری برخوردارند. کاربران باید درمورد دامنه های com که تجاری هستند بسیار دقت نمایند. این دامنه برای اهداف تجاری استفاده می شود. دامنه net به سایتهای غیر انتفاعی داده میشود اینکه تا چه اندازه این شرکتهای غیر انتفاعی اطلاعات صحیح ارائه میدهند خود جای بحث دارد. علی رغم تقسیم بندی رایجی که از دامنه ها به عمل می آید در عمل نظارت چندانی بر دامنه های ذکر شده وجود ندارد و با پرداخت مبالغ ناچیزی دامنه ها واگذار



می گردد. لذا اگر چه باید در بررسی دامنه و URL صفحه بازیابی شده دقت نمود اما این معیار به تنهایی معیار مناسبی برای رد یا تأیید اطلاعات بازیابی شده نیست.

### ۹-۱ تعیین میزان پوشش

اطلاعات ارائه شده در صفحات وب گاه بسیار عمیق و گاهی سطحی است. گاه تنها به ارائه لیستی از مطالب تخصصی اکتفا شده است بدون آنکه به دانش عمیق مربوط به آن اشاره ای گردد و گاهی ریزترین مسئله مورد بحث فراوان قرار گرفته است. باید در نظر داشت که طرح سوال، جستجو و بازیابی اطلاعات متأثر از نوع و محتوای نیاز اطلاعاتی، نوع و سطح مخاطبان، دامنه کار یا فعالیتی که به واسطه اطلاعات قصد پوشش دادن به آن را دارند و بسیاری عوامل دیگر است. صفحات وب نیز متأثر از اهداف طراحان آن سطح پوششی خاصی از مخاطبان را مد نظر دارند و پشتیبانی می کنند.

تخمین دامنه اطلاعات مناسب برای جستجوگر می تواند به چندین روش انجام پذیرد یکی از این روشها مطالعه فهرست مندرجات یا نقشه سایت -در صورت وجود- است، روش دیگر مطالعه و بررسی توصیفی است که اغلب سایتها از خود ارائه می کنند. سایتهایی که اطلاعات عمیق و تخصصی تری دارند اغلب امکان جستجو را از طریق موتور جستجوی خود فراهم می آورند در این صورت با انجام جستجوی ساده می توان تا حدی به دامنه اطلاعات ارائه شده و نوع سازماندهی آن پی برد. وجود پیوندهای فرامتنی نیز اگر به درستی برقرار باشند بر عمق پوشش مطالب می افزايند.

در مورد برخی از سایتها که مقالات تمام متن را فراهم می آورند پوشش گذشته نگر نیز اهمیت می یابد. تعیین پوشش همواره بستگی به سطح دانش کاربر و نوع نیاز وی دارد.

### ۱۰-۱ ارزیابی شهرت و اعتبار

شناسایی صاحبان اندیشه، شهرت و اعتبار آنها در ارزیابی صفحات وب از ضروریات است پس:  
- حامی صفحه [۴] یا فردی که مسئولیت حقوقی و معنوی سایت را عهده دار است مشخص نمایید. کاربران باید بتوانند ناشر یا حامی صفحه وب را شناسی کنند به گونه ای که ارزشها، اندیشه و اهداف وی مشخص شود. این بررسی با مراجعه به بخش هایی از صفحه که با "About Us" or "Mission" مشخص شده اند انجام می شود. حامی صفحه وب باید علاوه بر شماره تماس و آدرس پست الکترونیکی به صورت متمایز مشخص شده باشد (Smith, ۲۰۰۱).

- پدید آور را مشخص کنید.

گاه کاربران در اطلاعات بازیابی شده آنقدر غرق می شوند که پدید آورنده اثر و اطلاعات زمینه ای وی را از یاد می برند و به آن توجهی نمی کنند. در مورد وب سایت سازمانها و بسیاری از سایتهای دیگر امکان دارد که اطلاعات مربوط به پدید آورنده مطالب ذکر نشده باشد در این صورت اگر امکان آن



وجود داشته باشد با ایجاد ارتباط با مدیر سایت می توان [۵] اطلاعات مفیدی به دست آورد به علاوه اطلاعاتی که سازمانها تحت نام تنالگانی خود ارائه می کنند اغلب تحت نظارت کارمندان و متخصصان موضوعی است و در صورت معتبر بودن آن سازمان مشکلی در مورد عدم شناخت پدید آور نخواهد بود. در عین حال مشخص کردن صاحب اندیشه در هر سایتی نشان دهنده دقت نظر و احترامی است که طراحان آن به تفکر انتقادی مخاطبان خویش دارند (Smith, 2001).

در صورتیکه اطلاعات مربوط به پدیدآور مشخص باشد باید دقت داشته باشید که توضیحات کامل باشد و تصویر مناسبی از وضعیت مسئولیت مطالب منتشر شده را مشخص نماید. ارائه آدرس پست الکترونیکی نویسنده مطلبی که مشخصات حرفه ای و تخصصی وی را نمایان نسازد قابل اعتنا نیست. در اینصورت باید با نویسنده مطلب ارتباط برقرار کرد و مشخصات لازم را از او دریافت نمود. (Barker, 2004)

## ۱-۱۱ تعیین صحت

گاه اطلاعات ارائه شده در محیط مجازی وب نه تنها صحیح نیست که گمراه کننده و در مواردی مثل اطلاعات پزشکی خطرناک نیز هست لذا در ارزیابی اطلاعات باید بر این جنبه بسیار دقت داشت. تعیین میزان صحت اطلاعات بازایی شده مهمترین بخش ارزیابی انتقادی اطلاعات است. اغلب کاربران از روی ظاهر سایت و نوع طراحی آن به ارزیابی می پردازند در حالیکه صرف طراحی مناسب و زیبای سایت دلیلی بر صحت محتوای اطلاعاتی آن نیست. اشتباهات تایپی و گرامری گاه محتوا را قلب می کنند و اطلاعات گمراه کننده ای را منتقل می سازند. پدید آور مطالبی که محتوایی قابل اعتماد داشته باشند کلامش را با استنادها اعتبار می بخشند در صورتیکه مقاله ای علمی فاقد استناد و منابع باشد نباید محتوای اطلاعاتی آن را زیاد معتبر و جدی قلمداد کرد. در مورد سایتها نیز چنین است. اطلاعات ارائه شده در صفحه ای اینترنتی چنانچه با پیوندها مسیری را برای کشف منابع و استنادها فراهم آورد بر اعتبار خویش افزوده است (Smith, 2001).

## ۱-۱۲ روزآمدی

با توجه به رشد روز افزون علم و کوتاه شدن دوره حیات علمی یک اندیشه، لازم است از روزآمدی اطلاعات ارائه شده اطمینان حاصل گردد. توجه به تاریخ انتشار یک صفحه و تاریخ بازبینی آن که اغلب در پائین یا بالای صفحه ظاهر می شود گامی جهت ارزیابی روزآمدی صفحه و اطلاعات آن است. الزاما تاریخ انتشار صفحه و یا تاریخ روزآمدسازی سایت به تنهایی ملاک روزآمدی اطلاعات نیست، فعال نبودن پیوندها نشانی از عدم پویایی و روزآمدسازی سایت است. به علاوه منابع مورد استفاده در متن نیز باید مورد توجه قرار گیرد در صورت کهنه بودن منابع مورد استناد، باید در روزآمدی مطالب ارائه شده تردید کرد. توجه به این مسئله اهمیت دارد که به صرف کهنه بودن منابع مورد استناد نمی توان خط

بطلانی بر محتوای اطلاعاتی، اطلاعات بازایی شده کشید بلکه باید همزمان به متن آن علم یا رشته، طول عمر مفید اطلاعات در آن موضوع و بار محتوایی مطالب نیز توجه داشت.

### ۱-۱۳ قابلیت دسترسی پذیری

- دسترسی پذیری در شبکه

دسترسی به منابع اطلاعاتی در اینترنت ممکن است به شیوه های مختلفی محدود شود (از جمله زبان منبع، نیاز به ثبت نام، نیاز به اثبات صلاحیت یا عضویت در یک سازمان خاص) (خادمیان، ۱۳۸۲: ۱۰۲) دسترسی پذیری بزرگترین عاملی است که مراجعان کتابخانه ها را به پای میز کامپیوتر کشانده. محیط مجازی وب محدودیت زمانی و مکانی کاربران را از میان برداشته است و با ارائه انبوهی از اطلاعات آنان را در ظاهر از کتابخانه هایی نیاز ساخته است. اما کاربران غالباً فراموش می کنند که اغلب اطلاعات علمی و مستند در پایگاهها ارائه شده و دسترسی به آنها برای همه ممکن نیست.

از سویی کتابخانه ها و مراکز اطلاع رسانی به طراحی وب سایتهای خود پرداخته و دسترسی پذیری اطلاعات را از طریق شبکه ها و پایگاههای پیوسته نیز فراهم آورده اند. حرکت به سوی شبکه ها و کتابخانه های مجازی است. در چنین محیطی که کاربران در معرض انواع اطلاعات قرار گرفته اند و امکان استفاده از اطلاعات ضعیف تر به علت سهولت بازایی و دسترسی پذیری بیشتر وجود دارد باید به کاربران آموخت که کتابخانه ها نیز دسترسی به برخی از منابع اطلاعاتی را فراهم نموده اند و معیارهای ارزیابی منسجمی بر این انتخابها و گزینش ها تاثیر داشته است.

- دسترسی پذیری در صفحات یا نحوه ارائه و نظم دهی اطلاعات در صفحه نحوه ارائه اطلاعات در صفحات وب از اهمیت فراوانی برخوردار است. نوع سازماندهی اطلاعات، تعریف پیوند های فرامتنی، امکان استفاده از جستجو و مرور، میزان تعامل با کاربر و کاربر مدار بودن، سرعت بارگذاری اطلاعات، قابل فهم و ساختارمند بودن ارائه و به بیانی وجود سازماندهی در ارائه اطلاعات دسترسی پذیری را سهولت می بخشد و کاربران را از سردرگمی می رهااند.

نسبت به واژه هایی چون "additional sites," "related links," etc. حساس باشید. پیوند ها را بررسی کنید. ارائه ارجاعات کور، غیر فعال و یا نامربوط از اعتبار اطلاعات ارائه شده می کاهد (Barker, 2004).

این بخش از کار به امکانات تکنولوژیکی و مهارت در طراحی این نظامها بستگی دارد و توجه به آن در ارزیابی صفحات وب از اهمیت فراوانی برخوردار است.

#### ۱-۱۴ قیمت

هزینه لازم برای دسترسی به اطلاعاتی یکی از مواردی است که به هنگام استفاده از اطلاعات باید به آن توجه داشت. اطلاعات با ارزش پایگاههای اطلاعاتی معتبر دنیا به صورت اشتراکی ارائه می شوند. گاهی استفاده از چکیده و اطلاعات کتابشناختی رایگان است اما برای دریافت متن کامل نیاز به پرداخت وجه هستید. در نتیجه شما باید ارزش نسبی پرداخت هزینه ها را به محتوایی که دریافت خواهید نمود تخمین زده تصمیم گیری نمایید.

جهت خرید فایل word به سایت [www.kandooocn.com](http://www.kandooocn.com) مراجعه کنید  
یا با شماره های ۰۹۳۶۶۰۲۷۴۱۷ و ۰۹۳۶۶۴۰۶۸۵۷ و ۰۶۶۴۱۲۶۰-۵۱۱ تماس حاصل نمایید

## فصل دوم:

مقایسه متدهای قابل دسترس ارزیابی وب  
ومعلومات حاصل از کارایی متدهای ارزیابی



بررسی کارایی متد های ارزیابی (uems) نشان می دهد که تشابه قابل توجهی بین معیارهای کاربردی در هر دو وب سایت قابل دسترس و کاربردی وجود دارد . به علاوه بسیاری از منافع و مضرات که در روش Uems وجود دارد در روش متدهای ارزیابی قابل دسترس (Aems) نیز می تواند نشان داده شود.

در این فصل موثرترین روش ارزیابی قابل دسترس مشخص شده تا دسترسی وب صورت گیرد. و به این نتیجه می رسیم که روش کامل (یعنی روشی که ترکیبی از ارزیابی اتوماتیک ، دستی و تست های قابل دسترس می باشد). بهترین روش است. به علاوه برای سازمان هایی با محدودیت های مالی و زمانی روش قابل دسترس که ترکیبی از ارزیابی اتوماتیک و دستی می باشد، بهترین روش ارزیابی است.

همچنین در این فصل در مورد معیارهای مطالعه که کارایی رهنمودهای محتویات وب در دسترس را بررسی می کند بحث شده است که پیشنهاداتی برای تحقیقات بیشتر ارائه می کند.

تیم برنزلی مدیر ومخترع شبکه گسترده جهانی می گوید: " قدرت وب در جهان عمومی است . دسترسی هر شخص به وب صرفنظر از معلولیت ها ، جنبه ای مهم از آن است."

سولیوان و ماتسن می گویند: آغاز به وجود آمدن شبکه گسترده جهانی (www) به عنوان پایگاه بی طرف و وسیله ای مستقل تصور می شد که راهی برای دستیابی اطلاعات بود. تعداد وب سایت های قابل دسترس لزوما در حال افزایش نیستند حتی ممکن است در طی سال های اخیر با افزایش گرافیک و انیمیشن هایی با اطلاعات غنی در حال کاهش باشند.

محققان ، شاغلان و کمپانی های IT شماری از متد ها و ابزارها را گسترش دادند تا با سازمانها در جهت دسترس ساختن وب سایت هایشان همکاری کنند. نظریه های متفاوتی در مورد بهترین متد ارزیابی وجود دارد. با این وجود نسبتا مطالعات کمی در مورد روش مقایسه ای صورت گرفته است. با بررسی شمار کمی از مقالات قابل دسترس به این نتیجه می رسیم که مقایسات دشوار است زیرا متد های متفاوت متغیر های مختلف را ارزیابی می کنند به این دلیل ممکن است صحت نتایج برخی مطالعات سؤال برانگیز شود. شباهت های زیادی بین UEMS, AEMS وجود دارد. به عنوان مثال کلا تست های قابل دسترس برای کاربران به وجود می آیند تا معیارهای مشابه را در آزمایشات کارایی مانند اتمام وظیفه ، رضایت و راندمان به کار برند.

در کل تکنیک های دستی واتوماتیک دسترسی وب سایت را با استفاده از رهنمود های محتویات وب در دسترس ارزیابی می کنند. در نتیجه احتمال دارد که معلومات حاصل از بررسی قبلی روش مقایسه ای UEMS وجود داشته باشد که برای AEMS بکار می رود. این مسئله سؤالات زیادی ایجاد می کند :

۱- آیا رابطه ای بین متد های ارزیابی کارایی و قابل دسترس وجود دارد ؟

۲- چه متد های ارزیابی قابل دسترس موجود هستند منافع و مضرات متد های متفاوت چیست ؟

۳- آیا در حقیقت تطبیق کلی رهنمود های محتویات وب قابل دسترس این معنی را می دهد که وب سایت برای افراد معلول در دسترس است؟

۴- بنابر این مؤثر ترین روش ارزیابی قابل دسترس جهت تعیین دسترسی حقیقی وب سایت ها چیست؟

## ۲- ادلایل تضمین دسترسی وب سایت ها

وب منبع اصلی برای موارد زیر می باشد:

- اخبار، اطلاعات، تجارت، برنامه تفریحی
- آموزش کلاسی، یادگیری دوره ای
- کاریابی، ارتباط کارگاهی
- مشارکت ارتباطی، خدمات دولتی

وب جانشین منابع مرسوم اطلاعاتی و ارتباطی از قبیل:

- مدارس، کتابخانه ها، صنعت چاپ، مجموعه کارگاه
  - برخی منابع مرسوم قابل دسترس برخی غیر قابل دسترس بودند
- در نتیجه، احتیاج روز افزونی به منظور تضمین اینکه شبکه گسترده جهانی دستیابی بی سابقه ای به اطلاعات برای افراد معلول ارائه می دهد وجود دارد.
- همانطور که بحث شد، سه دلیل اساسی برای اینکه چرا سازمانها باید وب سایت شان را قابل دسترس بگذارند وجود دارد:

علل اجتماعی - طبق موازین اجتماعی سازمانها موظف هستند مسئول باشند و برای برطرف کردن تبعیض به منظور ارتقاء حقوق بشر تلاش کنند. این موضوع ساخت وب سایت های در دسترس برای تمام کاربران شامل افراد را ایجاب می کند.

علل تجاری/اقتصادی - مفهوم اقتصاد برای سازمان ها باعث نمی شود ده درصد از کاربران پتانسیل که شاید معلول باشند را از خود فراری می دهد. داشتن سایت غیر قابل دسترس ممکن است منجر شود که کاربران معلول برای سایت های رقابتی یا کلید زنی به سمت کانال هایی با هزینه بیشتر مراکز مکالمه ای و انشعابات وسیع کشیده شوند.

انجمن تشخیص افراد ناتوان در استرالیا افراد معلول را حمایت می کند که احتیاج است سازمانها دسترسی مشابه برای افراد ناتوان در منطقه اطلاعات و خدمات online جایی که منطقی می تواند مهیا شود ارائه دهد. سازمان هایی که در استرالیا در ساخت دسترسی سایت ها موفق نمی شوند ریسک عمل قانونی شامل مخالفت کمیته سازمانی سیدنی به منظور بازیهای المپیک را ابراز می کند.

## ۲-۲ منابعی که به دسترسی سایت کمک می کنند

کنسرسیوم شبکه گسترده جهانی معمولاً به W3C مراجعه می شود :

- یک کنسرسیوم بین المللی و تغذیه ای با بیش از ۵۰۰ عضو
  - ارتقاء ارزیابی و عمل پذیری وب و داشتن تمرکز عمده بر جهان شمولی وب (W3C)
- W3C دارای چهار حوزه می باشد ، یکی از آنها قوه ابتکار WAI است که دولت و صنعت سرپرستی آن را بر عهده دارد . قوه ابتکار WAI با سه حوزه دیگر W3C همکاری کرده و بین المللی عمل می کند WAI. سه منطقه کاری دارد.

۱- تضمین اینکه تکنولوژی های وب در دسترس بودن را حمایت می کند.

۲- گسترش رهنمود برای دسترسی

۳- بهبود ابزار جهت ارزیابی و اجرای دسترسی وب

۴- گسترش و توسعه مطالب به منظور آموزش و پیشرفت

۵- همکاری با تحقیق و توسعه

## ۲-۳ مبحث

این بخش به سه زیر بخش تقسیم شده است .بخش اول کارایی متد های ارزیابی و منافع و مضرات و پیامد معلومات کاربردی برای متد های قابل دسترس را بررسی می کند.بخش دوم منافع و مضرات متد های ارزیابی در دسترس را نشان می دهد . بخش سوم صحت و سود مندی را بررسی می کند . و مؤثر ترین متد ارزیابی در دسترس یا ترکیبی از تکنیک ها را برای تعیین درستی وب قابل دسترس مورد بررسی قرار می دهد .

الف - مقایسه کارایی متد های ارزیابی و معلوماتی برای متد های ارزیابی در دسترس  
مباحث و مطالعات متعلق به AEMS و اکثر مقالات در این زمینه جدید می باشند .متقابلاً ، همانطور که گری وسالزمن بیان کرده ، کارایی متدهای ارزیابی حاصل ، از اهمیت اصلی رابطه انسان - کامپیوتر محققان و متخصصان از دهه اولیه ۱۹۹۰ بوده است و مطالعات بی شماری بر روی مقایسه تأثیر این متد ها انجام گرفته است.

بر طبق ارتباط مقایسه ای AEMS که ضوابط کاربردی هستند UEMS و معیارهای کاربردی واقعی ارزیابی می شوند .

ب - معیارهای کاربردی در زمان مقایسه کارایی متد های ارزیابی

دامنه متغیر معیارهای مختلف در بررسی قبلی بکار رفته است به منظور اینکه کارایی تکنیک های ارزیابی را مقایسه کند که شامل موارد زیر می باشد :

- توانایی بیان کردن مسائل – اعداد و مسائل جدی تعیین شده
- صحت و کیفیت نتایج
- تأثیر هزینه ، کار و زمان
- سودمندی نتایج و نحوه بکار بردن نتایج – توانایی سازندگان برای درک و تحقق تغییرات مورد

احتیاج

- عمومیت نتایج – توسعه نتایجی که نماینده تمام کاربران هستند
- طبق موارد بحث شده تمام این معیارها می تواند در متد های قابل دسترس به کار گرفته شود .
- پ - معیارهایی که برای کارایی و قابلیت دسترسی استفاده می شوند
- گری تعریف ضعیف معیارهای کاربردی در مطالعات مقایسه ای اولیه UEMS را مورد انتقاد قرار داده است و آشکارا شرح می دهد هر چند چیزی ارزیابی می شود اما دور از ذهن است که این معیارها واقعا حساسیت به کارایی را منعکس می کنند .

معیار های عرضی که برای هر متد به کار گرفته شده در جدول یک به همراه روش های ارزیابی و مطالعاتی که استفاده از معیارها را مستند می سازند خلاصه شده اند . مقایسه ای از معیارهای به کار گرفته شده برای ارزیابی قابل دسترس و کارایی به صراحت نشان می دهد که تشابه قابل توجهی بین UEMS, AEMS وجود دارد.



جدول ۱-۲ - مقایسه روش ارزیابی و معیار های به کار گرفته شده روش کارایی و قابل دسترس

Table 1 - Comparison of evaluation approach and measures used for accessibility and usability.

Usability Evaluation		Accessibility Evaluation	
Measure	Study / UEM(s)	Measure	Study / AEM(s)
Number of usability problems and severity	<ul style="list-style-type: none"> <li>• Sullivan &amp; Matson (2000) / Automated (LIFT online)</li> <li>• Lindgaard (1999) / Heuristic evaluation</li> <li>• Law &amp; Hvannberg (2002) / Heuristic evaluation</li> <li>• Doubleday et al (1997) / Heuristic evaluation</li> </ul>	Number of accessibility problems / of W3C compliance issues and priority	<ul style="list-style-type: none"> <li>• Sullivan &amp; Matson (2000) / Automated and manual</li> <li>• Cooper &amp; Rejmer (2001) / Automated (Bobby)</li> <li>• Sloan (2002) / Automated and manual</li> <li>• Zaphiris et al (2001) / Automated (Bobby)</li> </ul>
Task completion	Usability testing <ul style="list-style-type: none"> <li>• Law &amp; Hvannberg (2002) / Usability testing</li> </ul>	Task completion	<ul style="list-style-type: none"> <li>• Sloan (2002) / Usability and accessibility testing</li> <li>• Colwell &amp; Petrie (1999) / Accessibility testing</li> </ul>
User satisfaction, opinions and emotional expression	<ul style="list-style-type: none"> <li>• Lindgaard (1999) / Usability testing</li> <li>• Macleod et al (1997) / Usability testing</li> <li>• Law &amp; Hvannberg (2002) / Usability testing</li> <li>• Doubleday et al (1997) / Usability testing</li> </ul>	User satisfaction and opinions	<ul style="list-style-type: none"> <li>• Sloan (2002) / Usability and accessibility testing</li> <li>• Colwell &amp; Petrie (1999) / Accessibility testing</li> </ul>
Duration – time to finish task	(2002) / Usability testing <ul style="list-style-type: none"> <li>• Doubleday et al (1997) / Usability testing</li> </ul>	Duration – time to finish task	<ul style="list-style-type: none"> <li>• Sloan (2002) / Usability and accessibility testing</li> </ul>

ت - معلوماتی که از مطالعات مقایسه ای متد های ارزیابی کارایی بدست آمده است.  
کیلام و هلند استدلال کردند که استفاده از ارزیابی کارایی در ارزیابی قابل دسترس به کار می روند  
جدول یک نشان می دهد که تشابه قابل توجهی بین معیارهای به کار گرفته شده در روش های ارزیابی و  
قابل دسترس و کاربردی وجود دارد که منجر به این می شود که چه معلوماتی می تواند از مطالعات قبلی  
بکار گرفته شود تا بتوانیم پایگاه اطلاعاتی صنعتی موجود را تولید کنیم .

لاو هوان برگ نظریه اساسی را مهم جلوه دادند آنها بیان کردند که گرایش ضروری منافع ومضرات  
پیشنهادات تجاری را مشخص می کند و شدیداً از UEMS حمایت می کند . آنها از تحقیق خود ذکر  
کردند که روش های متفاوت ، دسته های مجزایی از مسائل کارایی را تعریف می کند و هیچ کدام از  
متد ها تمام مسائل را ندارند . به عقیده آنها در واقع متد های ارزیابی و کارایی یکدیگر را کامل می کنند

گری از روشی طرفداری می کند که کارایی را از طریق " معیارهای چند گانه همگرا " برای ارزیابی استفاده کنند . با استفاده از این نتایج در رابطه با UEMS,AEMS مشخص می تواند استدلال کند که هیچ متدی مناسب نیست و ترکیبی از AEMS می تواند نتایج بهتری به وجود آورد .

عقیده مهم دیگری در رابطه با معیارهای کارایی برای ارزیابی قابل دسترس می باشد این موضوع در کارگاه متخصصان در دسترس نقل می شود که روش های ارزیابی قابل دسترس اغلب که معیارهای راندمان و بازدهی محدود می شود در حالی که معمولاً تست های کارایی معیاری از رضایت کاربر را در بر می گیرد. رضایت کارگاه یعنی تست های قابل دسترس باید کارایی راندمان و رضایت کاربر را اندازه بگیرند معیار مشابه در ارزیابی های کارایی استفاده می شود .تعداد بیشمار از منافع و مضرات UEMS به خوبی در AEMS بکار گرفته شده است .به ویژه بسیاری از این منافع و مضرات در بررسی های متخصص وجود دارد /ارزیابی مشابه در منافع و مضرات AEMS های اتوماتیک و دستی شرکت دارند به طور مشابه ، منافع و مضرات موجود در تست کارایی همچنین در تست قابل دسترس دیده می شوند .استثاء مهم این است که تمام نتایج از تست قابل دسترس به دست آمده است این موضوع بدین دلیل است که شاید نتایج تست قابل دسترس به معلولیت کاربران خاص و ورژن خاص تکنولوژی کمکی که آنها به کار می برند وابسته باشد.

- بررسی متد های ارزیابی قابل دسترس و منافع و مضرات آنها

گستره مهمی از متد های مختلف ارزیابی قابل دسترس موجود است اما محققان آن را مهم جلوه دادند و این متد ها منافع و مضرات دارند .بخش زیر تکنیک های ارزیابی مختلف را خلاصه می کند که توسط چکیده ای از منافع و مضراتشان همانطور که در نوشته ها مستند شده دنبال می شود .

الف - اعتبار سنجی اتوماتیک و ابزار ارزیابی

امروزه شمار زیادی از ارزیابی اتوماتیک تجاری و رایگان در دسترس است .فهرست طولانی از این ابزار در وب سایت W3C خلاصه شده است.

سه مورد از ابزار عامیانه تر شامل :

- بابی <http://www.w3.org>
- ابزار ارزیابی صفحات وب شخصی را در مقابل راهنمایی های ایجاد شده شبکه گسترده جهانی، قوه ابتکار وب در دسترس را آزمایش می کند.
- سنجش بصری وب در دسترس (WAVE)

**جهت خرید فایل word به سایت [www.kandooocn.com](http://www.kandooocn.com) مراجعه کنید**  
**یا با شماره های ۰۹۳۶۶۰۲۷۴۱۷ و ۰۹۳۶۶۴۰۶۸۵۷ و ۰۶۶۴۱۲۶۰-۵۱۱ تماس حاصل نمایید**

ابزار ارزیابی رایگان بازمینی اتوماتیک را اجرا کرده و پتانسیل، نتایج داوری افراد را مهم جلوه می دهد.  
طبق موارد بحث شده پر کاربرد ترین و ساده ترین رمز گشایی حاصل از نتایج بایی است. این نتایج حاصل  
از قوه ابتکار می باشد. و در موسسه معلولان در کانون افراد ناتوان توسعه یافته است.

• شبکه کاربردی

این سایت دامنه تغییرات محصولات به منظور اجرای W3C، رهنمود های در دسترس و سایر رهنمود  
های کاربردی را مورد آزمایش و نظارت قرار می دهد.

جهت خرید فایل word به سایت [www.kandooocn.com](http://www.kandooocn.com) مراجعه کنید  
یا با شماره های ۰۹۳۶۶۰۲۷۴۱۷ و ۰۹۳۶۶۴۰۶۸۵۷ و ۰۶۶۴۱۲۶۰-۵۱۱ تماس حاصل نمایید

فصل سوم:

برنامه وب سایت



### ۳-۱ مدیریت برنامه وب سایت

تمامی برنامه های وب سایت ، چه در اینترنت و چه مربوط به یکی از بخشهای اینترنت کشوری همانطور که در مدیریت برنامه وب تعریف شده ، طراحی شده اند.

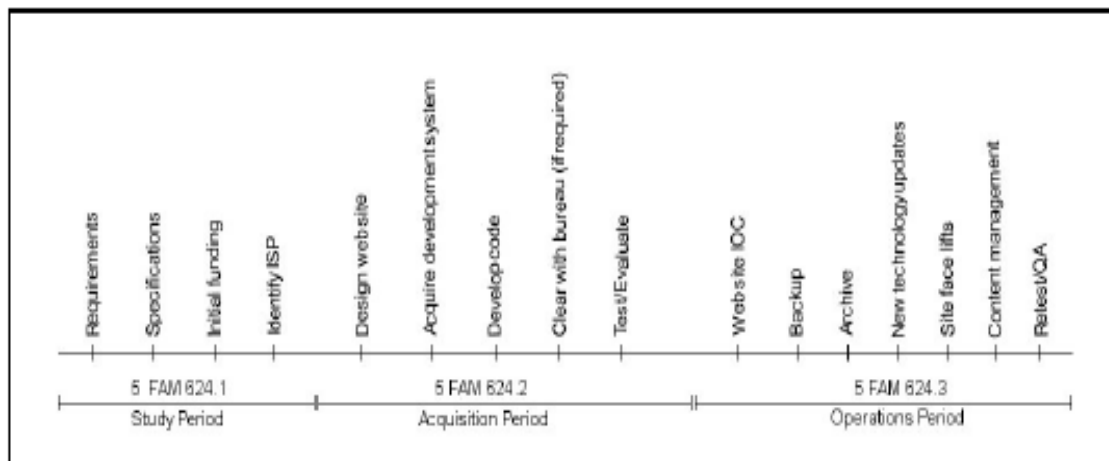
اداره برنامه های کشوری : برنامه های وب سایت دارای محتویات حساس و دیدنی زیادی می باشد که برای مدیریت محتوای مؤثر باید نظامی ظاهری داشته باشد و احتمالاً بیش از یکسال دایر باشد . طرح مدیریت کشوری (MSP) نیاز به برنامه هایی دارد این ویژگی ها باید با استفاده روش MSP اداره شود . میدان برنامه : هر وب سایت باید یک مدیر برنامه داشته باشد . اداراتی که دارای چندین وب سایت می باشند ممکن است برای هر سایت مدیر جداگانه ای انتخاب کنند و یا تمام وب سایت ها را به یک برنامه تحت نظر یک مدیر برنامه تبدیل کنند . وب سایت های پیچیده ممکن است به روش مدیریت تیمی نیاز داشته باشند ؛ اگرچه در نهایت مدیر برنامه عهده دار تمام وظایف و مسئولیت های تعریف شده می باشد.

#### ۳-۱-۱-۱ سنجش کارایی:

الف - اعضای تیم وب و سهام داران باید اهداف و مقاصد معین و محدود پذیری داشته باشند . اهداف و مقاصد چارچوبی را شکل می دهند که طبق آن وب سایت توسعه می یابد و به تعریف سهامداران برنامه نیز کمک می شود .

ب - اعضای تیم باید با استفاده از اهداف و مقاصد سنجش کارایی را برای ارزیابی مؤثر وب سایت تعریف کنند . این امور باید در طراحی برنامه وجود داشته باشد .

سیکل برنامه وب سایت : طبق سیکل برنامه تعریف شده ، سیکل طراحی برای تمام برنامه های IT کلیت دارد . این بخش به تعیین و توصیف دوره های خاص ، مراحل ، فعالیت ها و تأییدیه هایی می پردازد که برای توسعه وب سایت به کار می روند سه دوره مربوط به مدیریت منابع در شکل ۳-۱ نشان داده شده اند . (سیکل برنامه ای وب سایت)



## ۲-۳- ارزیابی آنچه در وب می بینید

وب جهانی مکانی است برای انجام تحقیقات در مورد موضوعات مختلف. اما قرار دادن مدارک و صفحات در وب راحت، کم هزینه یا رایگان، نامنظم و بدون نظارت می باشد. بزرگترین ارزشی که اینترنت برای اکثر جوامع به همراه دارد این است که مردم را قادر می سازد خودشان را معرفی کنند، یکدیگر را بیابند، نظراتشان را رد و بدل کنند، احتمالاً دوستان جهانی پیدا کنند. و از طریق لینک های hypertext در صفحات وب، خصوصیات و نظرات مردم را برای افرادی که وارد وب شده و روی آن کلیک می کند نمایش می دهد. از این رو توضیحی در مورد ارزیابی محتاطانه آنچه در وب یافت می شود، ذکر می شود. مسئولیت شما به عنوان خواننده این است که اعتبار، تألیف، مناسبت و درستی آنچه پیدا کرده اید را اثبات کنید. مدارک می توانند به صورت اختیاری و یا تصادفی به راحتی کپی و یا تحریف شوند. در وب جهانی عمومی ویراستاری برای تصحیح و بازگردانی و یا رد مطالب برای رسیدن به استاندارد معتبر برای چاپ وجود ندارد. بیشتر صفحاتی که در مورد موتورهای جستجوی وب وجود دارند خودبخود یا از طریق شرکتهای کوچک و بزرگ منتشر شده اند تا شما را برای خرید چیزی و یا باور یک عقیده تحریک کنند.

حتی در وب سایت های کتابخانه و دانشگاه ممکن است صفحاتی وجود داشته باشد که مؤسسات برای سرپرستی آن نکوشیده اند. وب باید عاری از چنین چیزی باشد. و شما اگر می خواهید به منظور تحقیقات مهم از آن استفاده کنید باید نسبت به درستی مطالب شکاک باشید و همه چیز را با تفکری منتقدانه مورد سؤال قرار دهید.

## ۳-۳ طراحی وب در دسترس چیست؟

به طور کلی، طراحی وب در دسترس مربوط به فلسفه و عملکرد طراحی وب می باشد. به طوری که صرف نظر از مکان اجرا و نوع تکنولوژی کامپیوتر مورد استفاده، توسط هر شخص قابل خواندن و هدایت می باشد.

طراحی وب در دسترس به طور رایج به بحث در مورد افراد ناتوان می پردازد. چون اگر اصول طراحی وب در دسترس به طور رایج به بحث در مورد افراد ناتوان می پردازد. چون اگر اصول طراحی وب در دسترس اجرا نشود این گروه احتمالاً آسیب می بیند. عدم موفقیت در پیشبرد این اصول دسترسی به صفحات وب را برای افراد ناتوان مشکل و یا غیر ممکن می سازد. طراحی دسترسی به محیط فیزیکی و طراحی دسترسی به وب جهان گستر شباهت های زیادی دارند. دسترسی ساختمانها و جنبه های دیگر محیط فیزیکی با طراحی محتاطانه و توجه به جزئیات بدست می آید. به علاوه با افزودن ویژگی های در دسترس در پایان فرایند طراحی. به همین ترتیب، ایجاد صفحات وب در دسترس بخش کاملی از فلسفه طراحی وب می باشد و ویژگی های دسترسی باید با تمام جنبه های فرآیند طراحی یکی شود.

به منظور دسترسی آزمون نیز باید با رژیم تستس تمام کاربران ترکیب شود و هرگز نباید حادثه ای مجزا که بعد از تست کاربران دیگر رخ می دهد در نظر گرفته شود بنا بر این طراحی دسترسی همانند موضوع استراتژیکی کاملاً تکنیکی می باشد.

در دسترسی نیاز نیست که تمام صفحات محدود به متن ساده ای باشد . صفحات جدیدتر و پیشرفته تر می توانند و نیز باید در دسترس باشند . به طور کلی ، این امر علاوه بر نیاز به پرهیز از طراحی جدید ، شامل تهیه جایگزینی برای ویژگی غیر قابل دسترس می باشد .

### ۳-۴ دسترسی برای وب

سازمان های دولتی ، تجاری و آموزشی استرالیا به منظور قابلیت دسترسی افراد زیادی به اطلاعات و دیگر خدمات به موقع و هزینه مؤثر به طور روز افزون از وب جهان گستر استفاده می کنند.دسترسی به اطلاعات و خدمات به شکل الکترونیکی از طریق وب جهان گستر عامل مهمی جهت دسترسی برابر برای افراد ناتوان بوده و دسترسی وسیع تر ، ارزان تر و سریع تری را نسبت به استفاده از دیگر فرمت های ممکن فراهم می کند .برای مثال :افراد ناشنوا یا دارای اختلالات شنوایی می توانند دسترسی وسیع تری به عنوان یا متن مواد شنیداری داشته باشد به علاوه افراد ناتوان ، به طور متوسط درآمد کمتری نسبت به افراد دیگر جامعه دارند و در نتیجه ممکن است به نرم افزارها و تجهیزات دسترسی نداشته باشند.براساس اطلاعات کارشناسی موجود ،روشن است که از نظر تکنیکی دسترسی به صفحات وب جهان گستر برای افراد ناتوان امکان پذیر می باشد.

جهت خرید فایل word به سایت [www.kandooocn.com](http://www.kandooocn.com) مراجعه کنید  
یا با شماره های ۰۹۳۶۶۰۲۷۴۱۷ و ۰۹۳۶۶۴۰۶۸۵۷ و ۰۶۶۴۱۲۶۰-۵۱۱ تماس حاصل نمایید

فصل چهارم:

نحوه کار برنامه های کاربردی وب



برنامه های کاربردی وب دسترسی پویای وب پیچ های مبتنی بر HTML از طریق یک مرورگر وب را ایجاد می کند. از آنجا که برنامه های کاربردی Web Object شیء گرا بوده و به زبان جاوا نوشته می شود، برنامه کاربردی شما Web page هایی را با ایجاد نمونه های شیء ها به نام مؤلفه های وب به وجود می آورد.

مؤلفه وب ترکیبی از زیرمجموعه های Wocomponent و قالب HTML جاوا است. مؤلفه های وب همه مؤلفه ها و عناصر استاندارد HTML شامل انیمیشن Flash، فیلم های Quicktime، برنامه های Java script و اپلت های جاوا را در برمی گیرد. مؤلفه های وب همچنین از کاسکاد ورقه های تعاریف (CSS) پشتیبانی می کند.

با افزودن عناصر خاص WebObject با همت های HTML - به نام عناصر پویا- به مؤلفه های وب خود، محتوای پویا را به وب پیچ های خود اضافه کنید. برخی عناصر پویا فاقد همت های HTML بوده و صرفاً در کنترل ایجاد محتوا به کار می رود. به عنوان مثال، محتوایی که شرطی یا تکراری است. هنگام پاسخ دهی به درخواست های مرورگر کلانیت، محتواهای پویا به HTML ایستا ترجمه و تبدیل می شود. می توانید در ساخت مؤلفه های وب یا از WebObject Builder یا Direct to Web استفاده کنید. WebObject Builder یک ابزار گرافیکی در خلق مؤلفه های وب بوده و عناصر پویا را به متغیرها و متدهای برنامه کاربردی شما محدود می کند. Direct to Web یک ابزار ساخت سریع نمونه بوده که برنامه کاربردی در حال کار وب را از مدل E0 ارائه شده خلق می کند. برای تغییر محتوای مؤلفه های Direct to Web از Web Assistant استفاده نمایید. همچنین می توانید صفحات را فریز کنید- مؤلفه های وب را خلق کرده و به پروژه خود اضافه کنید- و آنها را با استفاده از Web Object Builder اصلاح نمایید.

باید قبل از معمول سازی برنامه کاربردی وب خود، درک بنیادینی از معماری یک برنامه کاربردی داشته باشید. این قسمت معماری برنامه های کاربردی وب را تشریح کرده و چگونگی کار عناصر پویا را در زمینه حلقه درخواست- پاسخ برنامه کاربردی را توضیح می دهد. هنگام پردازش یک درخواست و ایجاد صفحه پاسخ، شرح خلاصه ای از ترتیب متدها استنتاج می شود. این قسمت همچنین چگونگی کاربک تراک (پیمایش معکوس) Web Object ها را توضیح می دهد.

این سند چگونگی استفاده از ابزارهای مختلف Web Object را توضیح نمی دهد. در مورد مراحل استنتاجی در خلق قالب ها و ساخت عناصر پویا، راهنمای کاربر Web Object Boilder را مطالعه کنید. در مورد چگونگی استفاده از Direct to web, Web Objects Direct to web Guid (هدایت شیء های وب به راهنمای وب) را مطالعه نمایید.

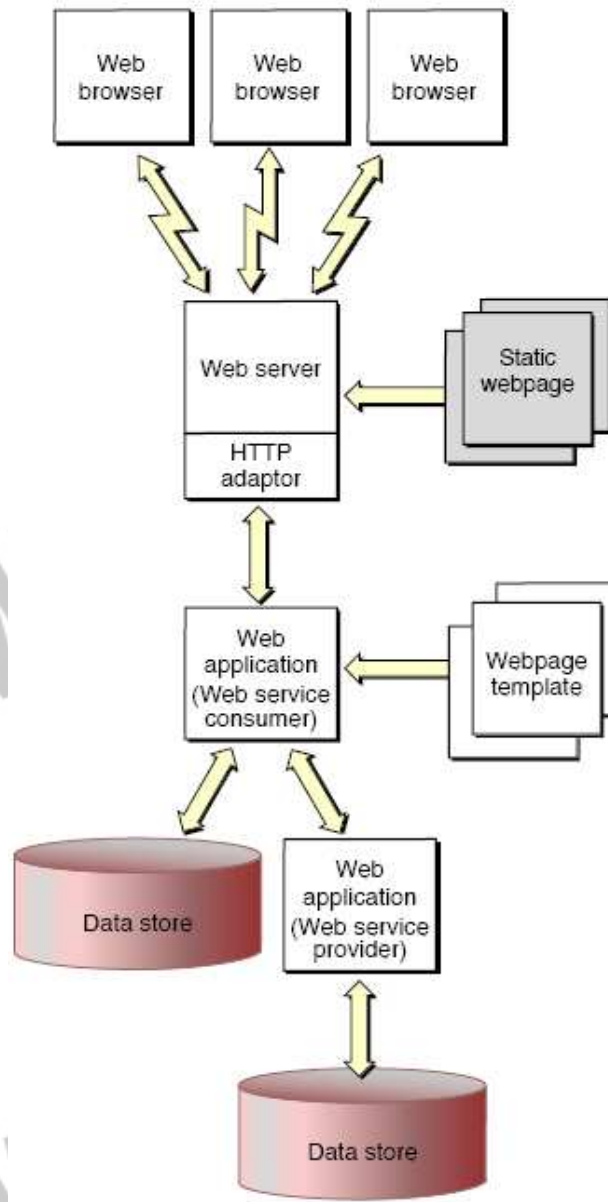
## ۴-۱ معماری برنامه کاربردی

نه تنها برنامه وب شما محتوای پویایی ایجاد می کند بلکه می توانید قالب ها را به کاربر ارائه داده که محتوای مؤلف را برای آنها تأیید می کند. داده های ورودی را از کاربرانی بدست می آورید که از قالب های HTML، دکمه ها و دیگر عناصر پویا استفاده می نمایند. مرتبط سازی عناصر قالب به متغیرها و متدهای مؤلفه وب شما شبیه محدود کردن دیگر عناصر پویاست که صرفاً محتوا را نمایش می دهد. با قرار دادن عناصر پویا در یک عنصر استاندارد قالب مؤلفه وب خود، قالب ها را خلق کنید. مؤلفه وب HTML را ایجاد می کند که مرورگرهای وب می توانند آن را تغییر کرده و نمایش دهد. این فرآیند از ترجمه دیتای ثبتي کاربر یا انتخاب ها به متغیرهایی در برنامه شما تشکیل می شود. اگر مؤلفه های وب را برنامه نویسی می کنید، این برنامه نویسی به درک چگونگی پردازش ورودی کاربر توسط برنامه های وب کمک می کند.

برنامه های Web Object مبتنی بر رویداد (event driven) هستند اما به جای پاسخ دهی به رویدادهای ماوس و کیبورد به درخواست های HTML (پروتکل انتقال ها پیرتکست) پاسخ می دهند. برنامه درخواست عمل HTML را دریافت می کند، به آن پاسخ می دهد و سپس منتظر درخواست بعدی می شود. پاسخ دهی برنامه به درخواست ها ادامه می یابد تا اینکه درخواست ها تمام شود. حلقه اصلی به کار گیرنده این درخواست ها حلقه درخواست-پاسخ، WebObject محتوای عناصر پویا را پر می کنند. اطلاعات انتشاری برنامه های شما در دیتابیس یا دیگر وسایل ذخیره سازی دیتا مقیم شده یا می توان آن را هنگام دسترسی به صفحه ایجاد کرد. صفحات هم بسیار محاوره ای و فعل و انفعالی هستند- شما می توانید کاملاً روش را مشخص کنید که کاربر از طریق آنها هدایت شده و دیتاهایی که آنها مشاهده و اصلاح می کنند.

شکل یک وب سایت مبتنی بر Web Object را نمایش می دهد. مجدداً درخواست (در قالب URL) از مرورگر وب به وجود می آید. مرور وب کشف می کند که درخواست باید توسط برنامه کاربردی Web Object به کار رفته و درخواست را به آداپتور HTTP بفرستد. آداپتور درخواست وارده را به شکلی پکیج می کند که برنامه Web Object بتواند آن را درک کرده و به برنامه کاربردی فوروارد کند. براساس مؤلفه های وب تعریف شما و دیتای مربوطه از انبار دیتا، برنامه وب پیچی را ایجاد می کند که از طریق آداپتور به سرور وب منتقل می شود. سرور وب صفحه را به مرورگر وب می فرستد که آن را ارائه می دهد.

به این نوع از برنامه کاربردی Web Object برنامه Web گفته می شود زیرا نتیجه آن ایجاد یک سری وب پیچ های HTML پویا می باشد.



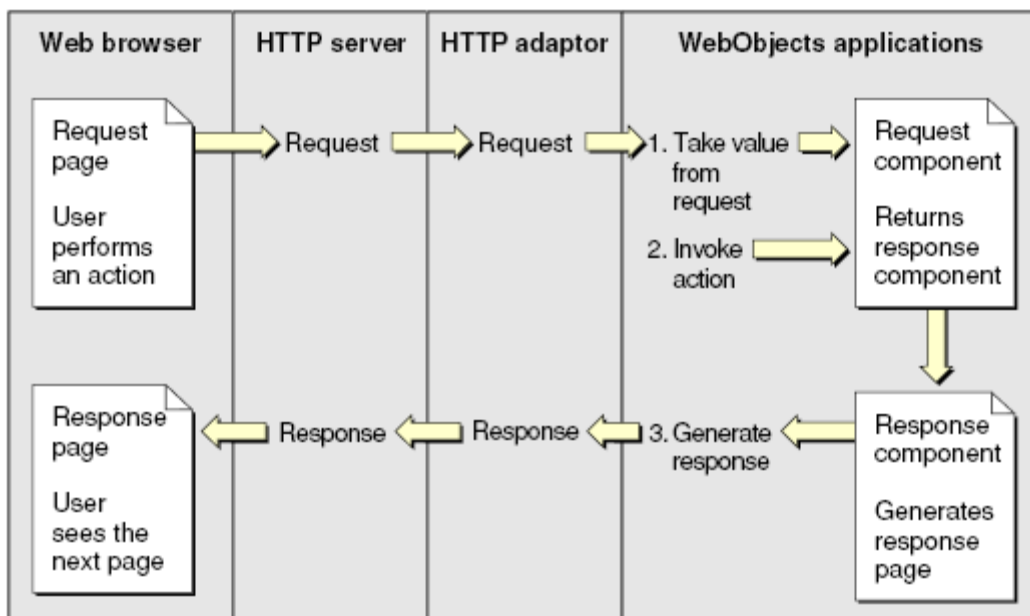
شکل ۴-۱ انتشار پویای وب

## ۴-۲ حلقه درخواست - پاسخ

هر عمل اتخاذ شده از سوی کاربر از طریق سرور وب و آداپتور Web Object با برنامه کاربردی شما ارتباط برقرار می کند. همه جزئیات مربوط به عمل کاربر - محتوای فیلدهای تکست، وضعیت چک باکس ها و دکمه های رادیویی و انتخاب در منوهای pap-up همین طور اطلاعاتی پیرامون session و دکمه یا لینک فعال شده در درخواست HTTP رمزگذاری می شود.

درخواست با عمل آداپتور Web Object و رفتار پیش فرض برنامه رمزگشایی می شود. این فرآیند رمزگشایی، که در ایجاد صفحه پاسخ به حداکثر خود رسیده تا به مرورگر وب بازگشت داده شود - حلقه

درخواست- پاسخ را می سازد. شکل ۴-۲ ترتیب پیام های استنتاجی هنگام پردازش درخواست را نشان میدهد.



شکل ۴-۲ حلقه درخواست

Web Objects دو مدل پردازش درخواست دارد، عمل هدایت و عمل مؤلفه.  
- مدل عمل مؤلفه به شما اجازه می دهد وضعیت را در برنامه حفظ کنید؛ از این رو، به شیء های session نیاز داشته و از آن استفاده می کند. به طور پیش فرض، برنامه های وب از این مدل استفاده می کنند.

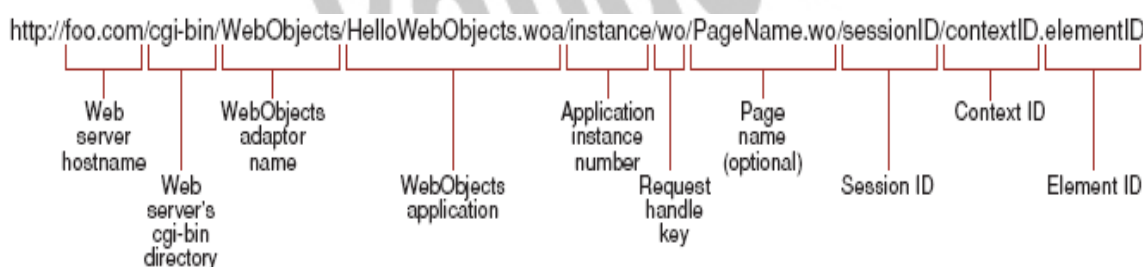
- مدل عمل مستقیم توسط برنامه هایی به کار می روند که به مدیریت وضعیت- به عنوان مثال، موتورهای جستجو، کاتالوگ های تولید، کتابخانه های پرونده و انتشار پویا- نیاز ندارد. برنامه های استفاده کننده این مدل به طور پیش فرض فاقد شیء های session هستند.

هنگام توسعه یک برنامه به یک مدل پردازش درخواست محدود نمی شوید برنامه ها مناسب ترین مدل را در پیاده سازی مشخصه های خاص به کار می گیرید. عمل های مؤلفه به طور کلی در برنامه های وب دارای مؤلفه های به هم پیوسته سودمند است. به هر حال، توانایی کنترل زیاد جریان برنامه را به کاربر ارائه نمی دهد. به عنوان مثال، کاربر نمی تواند مستقیماً متد تعریف شده در فایل سرس جاوای یک مؤلفه وب را اجرا کند. از طرف دیگر، عمل های هدایت (Directaction) در ارائه چنین دسترسی به کاربران بهترین است. به عنوان مثال، با به کارگیری URL مناسب کاربران می توانند شیوه های خاص یک برنامه را اجرا کنند.



### ۳-۴ URL های عمل مؤلفه

هنگامی که یک برنامه وب را آرایش داده و از طریق مرورگر وب به آن دسترسی دارید، URL نمایش مرورگر از فرمت خاصی برخوردار است که زمینه، session صفحه برنامه وب و حتی عنصر وب را مشخص می کند. شکل 3-4 قسمت های URL را نشان می دهد. URL همه اطلاعات ضروری برنامه کاربردی را در بر گرفته تا وضعیت مؤلفه های وب و session ایجاد شده برای کلانیت مشخص را بازسازی کند. لیست ۱ مثالی از URL عمل مؤلفه را نشان می دهد.



شکل ۳-۴ ساختار یک URL عمل مؤلفه

• لیست یک - مثالی از یک URL عمل مؤلفه

`http://foo.com:49663/cgi-bin/WebObjects/TimeDisplay.woa/wo/NDdW3uF2xRVjvbXUgRCVM/0.5`

### ۴-۴ پیام های حلقه درخواست - پاسخ

جدول 1-4 فازهای فرآیند درخواست - پاسخ را فهرست می کند. جدول 2-4 ترتیبی را نشان می دهد که متدهای موجود در آن استنتاج می شود. فرآیند به طور تفصیلی در "پردازش درخواست" و "ایجاد پاسخ" توضیح داده می شود. شیء های اولیه دریافت کننده پیام ها از حلقه درخواست - پاسخ عبارتند از شیء مؤلفه وب، session و برنامه شیء برنامه (application object) مثال برنامه ای است که زیر مجموعه WOApplication می باشد. یک session object مثالی از session بوده که session زیر مجموعه ای از WOApplication می باشد. هنگامی مثال Application خلق می شود که برنامه شما آغاز می شود و مثال session برای هر کاربر اولیه شکل می گیرد. توجه کنید که ممکن است session ها تایم اوت شوند. می توانید زمان تایم اوت را هنگام آرایش یک برنامه پیکربندی کنید.

اگر یکی از قالب های برنامه های وب در Xcode را هنگام ایجاد پروژه ای انتخاب کنید، کلاس های Application, session به طور اتوماتیک به پروژه شما اضافه می شود. در مورد چگونگی ایجاد پروژه Webobject Xcode قسمت "ایجاد پروژه ها" را بخوانید.

جدول ۴-۱: تایم لاین پردازش درخواست - پاسخ

Application	Session	Component
awake		
	awake	
		awake
takeValuesFromRequest		
	takeValuesFromRequest	
		takeValuesFromRequest
		Set methods invoked.
invokeAction		
	invokeAction	
		invokeAction
appendToResponse		
	appendToResponse	
		appendToResponse
		Accessor methods invoked. Response page generated.
		sleep
	sleep	
sleep		

## ۴-۵ پردازش درخواست

عمل پردازش درخواست در سه مرحله روی می دهد: بیداری، سنکرون، و عمل - Awake (بیداری) - هنگامی این مرحله اجرا می شود که Web object پیام های awakwe را به چندین شیء می فرستد. ضروری است که از منابع محدود در سیستم چند کاربری تا حد ممکن به طور سودمند استفاده می شود. برنامه ها صرفاً زمانی فعال هستند که کاری انجام دهند. یک سرور مجزا می تواند چندین برنامه یا نمونه های زیادی از همان برنامه را اجرا کند. نمونه های برنامه صرفاً حین پردازش درخواست ها فعال هستند برای کسب اطلاعات بیشتر قسمت "ایجاد پاسخ" را ببینید. ابتدا شیوه awake شیء برنامه، سپس شیوه awake شیء session و متد awake مولفه وب برای درخواست های مبتنی بر عمل مؤلفه استنتاج می شود. شما می توانید متد را در هر یک از کلاس های متناظر متعارف (Custora) کنید تا بر منطقی بیافزاید که ضروری است قبل از پردازش درخواست انجام شود ولو آنکه پیاده سازی های پیش فرض این متدهای awake کاری انجام ندهند، باید پیاده سازی کلاس بالا را قبل از اجرای منطق معمول استنتاج کنید بدینگونه که:

```
public void awake() {  
    super.awake();  
  
    /* Custom logic goes here. */  
}
```

سنکرون- در طی این مرحله متد take Values From Request استنتاج می شود که باعث شده مقادیر ثبتي در عناصر قالب توسط کاربر در متغیرهای نمونه متناظر کپی شود. در صورتی از این مرحله پرسش می شود که مؤلفه فاقد عناصر قالب باشد یا مقادیر عناصر قالب تغییر نکرده باشد. Web Object متد take Values From Request شیء برنامه را استنتاج می کند. سپس برنامه شیوه متناظر شیء session را استنتاج می کند که در عوض، متد مولفه وب (برای درخواست های مبتنی بر عمل مؤلفه) را استنتاج می نماید. مؤلفه هر یک از متد take Values From Request عنصر پویا را استنتاج می کند که باعث می شود عناصر قالب مقادیر درخواست را در اتصالات مناسب مؤلفه کپی کند. web objects از کد گذاری مقدار کلید- پیاده سازی شده توسط رابط NSKeyValue Coding در (com-webobjects.foundation) - استفاده کرده تا چگونگی تنظیم مقدار اتصال را مشخص کند. برای تنظیم کلیدی به نام key کد گذاری مقدار کلید برای متد موجود یا متغیر نمونه به ترتیب ذیل به نظر می رسد:

1. public void setKey()
2. private \_setKey()
3. \_key
4. key

عمل - طی این مرحله، عمل انتخاب کاربر به وسیله استنتاج متد invoke Action اجرا می شود.  
Webobject مانند شیوه take Values From Request شیوه invoke Action برنامه را استنتاج می کند.  
سپس، برنامه متد session را استنتاج کرده که در عوض متد مؤلفه مبتنی بر وب (برای درخواست های مبتنی بر عمل مؤلفه) را استنتاج می کند. سپس مؤلفه invoke Action را به هر یک از عناصر پویای خود می فرستد.  
هنگامی که متد invoke Action عنصر پویای راه انداز درخواست - به عنوان مثال، یک دکمه تأیید - استنتاج می شود، عنصر پویا پیام محدود را به صفت action خود می فرستد.

#### ۴-۶ ایجاد پاسخ

بعد از اینکه مقادیر قالب جمع آوری شده و متد عمل استنتاج می شود، برنامه صفحه پاسخ را خلق می کند. این مؤلفه وب برگشت داده شده توسط متد action است. فرآیند ایجاد پاسخ از دو فاز تشکیل می شود: پاسخ و خواب.

- پاسخ - صفحه پاسخ در طی این فاز به وجود می آید. همه متدهای appendtoReponse عنصر پویا استنتاج می شود بنابراین، محتوای خود را به وب پیچ رندر و ارائه شده می افزاید.  
Web object متد appendtoReponse برنامه را استنتاج می کند. سپس، برنامه متد session را استنتاج می کند که در عوض متد مؤلفه وب را استنتاج می کند. مؤلفه از طریق HTML خلق کننده محتوای صفحه خود پیش می رود. هنگامی که مؤلفه عنصر WEBOBJECT را می یابد، متد appendtoReponse خود را استنتاج می کند بنابراین مقادیر محدودیت های خود را به دست آورده و محتوای حاصل را به صفحه می افزاید. فرآیند به طور برگشتی ادامه می یابد تا اینکه کل صفحه پاسخ ایجاد شود.

مجدداً هنگامی Webobject از کد گذاری مقدار کلید استفاده می کند که ضروری بوده متغیر تنظیم یا قابل دسترس شود. هنگامی که مقدار کلیدی به نام key درخواست شود، کد گذاری مقدار کلید ابتدا



شیوه accessor را جستجو می کند. اگر شیوه ای پیدا نشود، به خود متغیر نمونه دسترسی پیدا می کند.  
ترتیبی که کدگذاری مقادیر کلیدی سعی می کند مقدار key را به دست آورد بدین گونه است:

1. public [...] getKey()
2. public [...] key()
3. private [...] \_getKey()
4. private [...] key()
5. [...] \_key
6. [...] key

sleep - (خواب) - هنگامی که فرآیند پاسخ کامل شود، متدهای sleep شیء های برنامه session و مؤلفه  
وب استنتاج می شود (ترتیبی که متدهای sleep شیء در آن فراخوانی می شود متضاد ترتیبی است که  
متدهای awake در آن در فاز بیداری استنتاج می شود) هنگام لغو متدهای sleep باید پیاده سازی سوپر کلاس  
را در انتهای متد قرار دهید. که در لیست ۲ نشان داده می شود. بعد از آنکه همه شیء های استنتاجی در  
فرآیند درخواست پاسخ به خواب رفتند، صفحه جدید به آداپتور webobject ارسال می شود.  
لیست ۲- لغو متدهای sleep

```
public void sleep() {  
    /* Custom logic goes here. */  
  
    super.sleep();  
}
```

#### ۴-۷ بک تراک کش

webobject از استفاده دکمه Back مرورگر وب با نگهداری کش صفحات اخیراً مشاهده شده در سرور  
پشتیبانی می کند. این فرایند back tracking نامیده می شود. یک کش به طور پیش فرض پیکربندی می  
شود تا ۳۰ صفحه در هر جلسه در برداشته باشد اما می توان آن را برای تأمین نیازهای شما بهبود بخشید.  
برای تغییر سایز پیش فرض کش، کد را به سازنده کلاس Application اضافه کنید. به عنوان مثال، برای  
تغییر اندازه کش صفحه به ۴۵ صفحه، این سطر کد را اضافه کنید:

```
setPageCacheSize(45);
```

هنگامی که صفحه پاسخ ایجاد می شود، صفحه پاسخ و اطلاعات وضعیت آن به کش اضافه می شود. به این طریق، هنگامی که کاربر دکمه Back مرورگر را کلیک می کند، webobject می تواند مؤلفه صحیح وب و وضعیت آن را نگهدارد.

برای کار صحیح بک تراک با دیتای پویا، کش خود مرورگر وب باید غیرفعال شود به طوری که همه درخواست های صفحه به سرور وب و بنابراین، برنامه شما می شود. شما می توانید این را با افزودن این کد به متد سازنده کلاس Application انجام دهید:

```
setPageRefreshOnBacktrackEnabled(true);
```

هنگامی که کش پر می شود، قدیمی ترین صفحه در آن dircard شده تا فضایی برای خود ذخیره سازی یک صفحه جدید ایجاد کند. هنگامی که بک تراک های کاربر از قدیمی ترین صفحه در کش می گذرد، webobject با صفحه خاص به کاربر هشدار می دهد.

برای کسب اطلاعات بیشتر پیرامون بک تراک، قسمت "مدیریت کش و بک تراک" را مطالعه کنید.

## ۴-۸ ایجاد پروژه ها

یک پروژه webobjects همه فایل های مورد نیاز ساخت و اجرای کاربردی شما را در برمی گیرد. در ایجاد پروژه جدید webobjects از Xcode استفاده کنید. در Xcode قالب مناسب پروژه webobjects را انتخاب نمایید و دستياری از طریق فرآیند ایجاد پروژه شما را راهنمایی می کند. انواع فایل های اضافه شده به پروژه Xcode و سازمان دهی شما به الگو و قالب انتخابی شما بستگی دارد. اگر چه ممکن است چار چوب ها، مقصدها و پیکربندی های ساخت اندکی متفاوت باشند ولی سازمان دهی برنامه های وب-برنامه هایی که محتوای پویای HTML ایجاد می کند - خیلی به هم شبیه هستند.

این پروژه چگونگی استفاده Xcode در خلق برنامه های وب را توضیح می دهد. این پروژه قالب های مختلفی را تشریح می کند، دستور العمل مرحله به مرحله ایجاد پروژه شما را ارائه می دهد، سازماندهی فایل های موجود در پروژه را توضیح می دهد، مقاصد خاص برنامه وب را شرح می دهد، و ترفندهایی برای ساخت و نصب برنامه شما را در برمی گیرد.

## ۴-۸-۱ انتخاب یک قالب

هنگامی که پروژه ای به زبان Xcode به وجود می آید، انتخاب قالب و الگوی مناسب webobjects در assistant ضروری است. قالب های خلق کننده یک برنامه کاربردی وب عبارتند از برنامه Direct to

web، برنامه Display Group، و برنامه webobject. همچنین می توانید از چارچوب webobjects استفاده کنید.

- اگر یک مدل EO دارید که قبلاً یا با EOModeler یا Xcode ایجاد شده و می خواهید یک نمونه سریعی بسازید، برنامه Direct to web را انتخاب کنید. این انتخاب خوبی برای برنامه نویسان جدید در webobjects به شمار می آید.

- اگر مدل EO دارید یا می خواهید یکی ایجاد کنید- یعنی می خواهید وب پیج های خود را با محتوایی از دیتابیس back-end (انتهای برگشتی) مشهور کنید- و می خواهید مؤلفه های متعارف وب را بسازید، برنامه Display Group را انتخاب کنید.

- اگر نمی خواهید از شیء های سازمانی استفاده کنید، برنامه webobjects را انتخاب کنید.  
- اگر می خواهید چارچوبی ایجاد کنید، webobjects framework را انتخاب کنید. به طور نمونه، این قالب را برای ایجاد چارچوب دارای منطق کاری خود- مدل EO و شیء های سازمانی خود- انتخاب کنید که می توان آن را دوباره در انواع دیگر برنامه ها نظیر webservice به کار برد. همچنین می توانید چارچوبی از مؤلفه های قابل استفاده مجدد وب را ایجاد کنید.

## ۴-۸-۲ ایجاد یک پروژه برنامه وب

هنگامی که پروژه ای را از قالبی خلق می کنید، Xcode Assistant با نمایش تعدادی از کادرها شما را از طریق فرآیند راهنمایی می کند. اولین کادرهای محدود برای همه گونه های برنامه های وب یکسان است. کادرهای بعدی می تواند بر اساس قالب انتخابی شما تغییر کند. تنظیمات پیش فرض Assistant در اکثریت برنامه ها کار می کند. به طور نمونه، صرفاً لازم است نام پروژه را وارد کرده و دکمه Next را کلیک کنید و در کادر نهایی، روی Finish کلیک کنید. این مراحل عمومی را برای خلق یک برنامه وب دنبال کنید. برای کسب جزئیاتی پیرامون استفاده الگوی Direct to web Guide webobjects، Direct to web Application را مطالعه کنید.

۱- Xcode واقع در Developer/Application را شروع کنید.

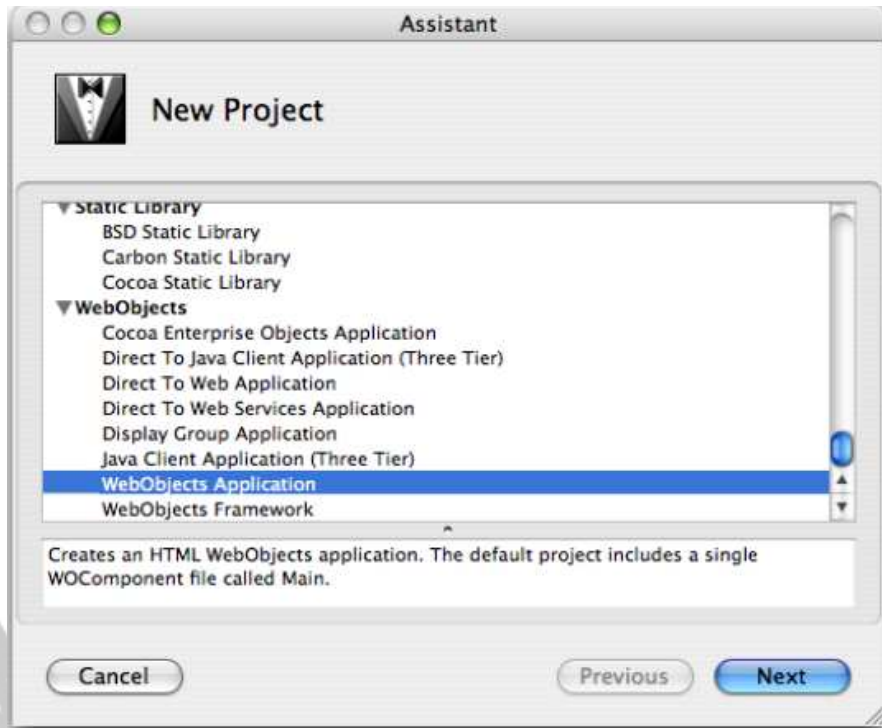
۲- منوی File > new perobject را انتخاب کنید.

پنل Assistant در حال نمایش کتبی از الگوها و قالب ها آشکار می شود.

۳- همانگونه که در شکل ۱ نشان داده می شود یکی از قالب ها و الگوهای webobjects را انتخاب کرده و Next را کلیک کنید.

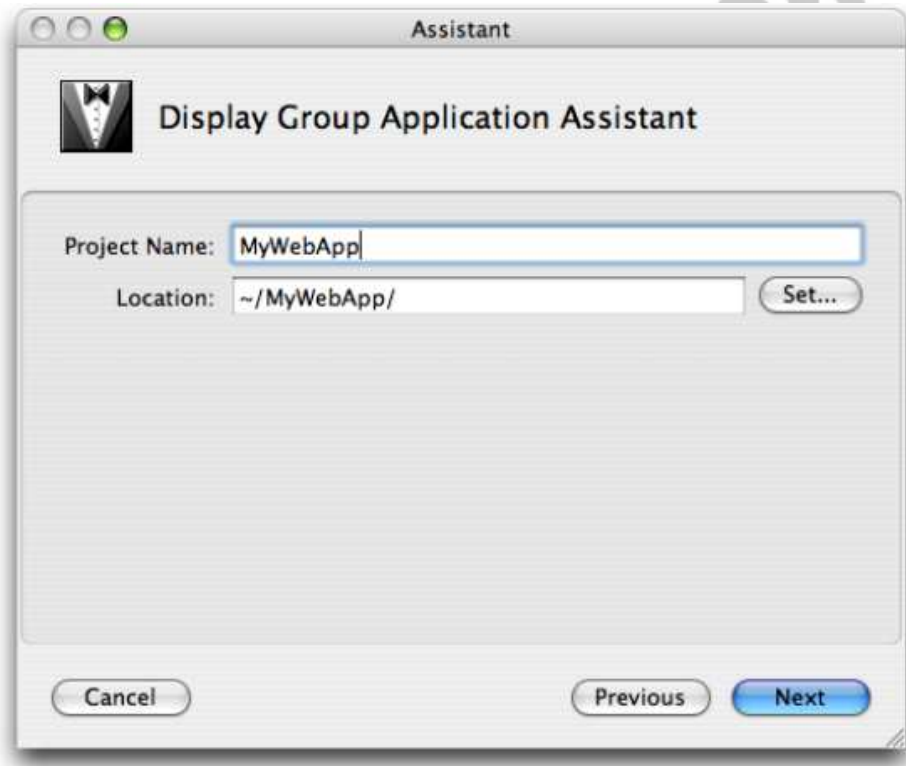
اگر مطمئن نیستید از چه قالبی استفاده می کنید، قسمت "انتخاب قالب" را مطالعه کنید.

جهت خرید فایل word به سایت [www.kandoocn.com](http://www.kandoocn.com) مراجعه کنید  
یا با شماره های ۰۹۳۶۶۰۲۷۴۱۷ و ۰۹۳۶۶۴۰۶۸۵۷ و ۰۵۱۱-۶۶۴۱۲۶۰ تماس حاصل نمایید



شکل ۴-۴: انتخاب یک قالب *webobject*.

۴-۵- نظیر شکل ۴-۵ نام پروژه و موقعیت را وارد کرده و Next را کلیک کنید. اگر الگوی *webobject* *Famenetwork* را کلیک کرده و به مراحل مانده پرش کنید.

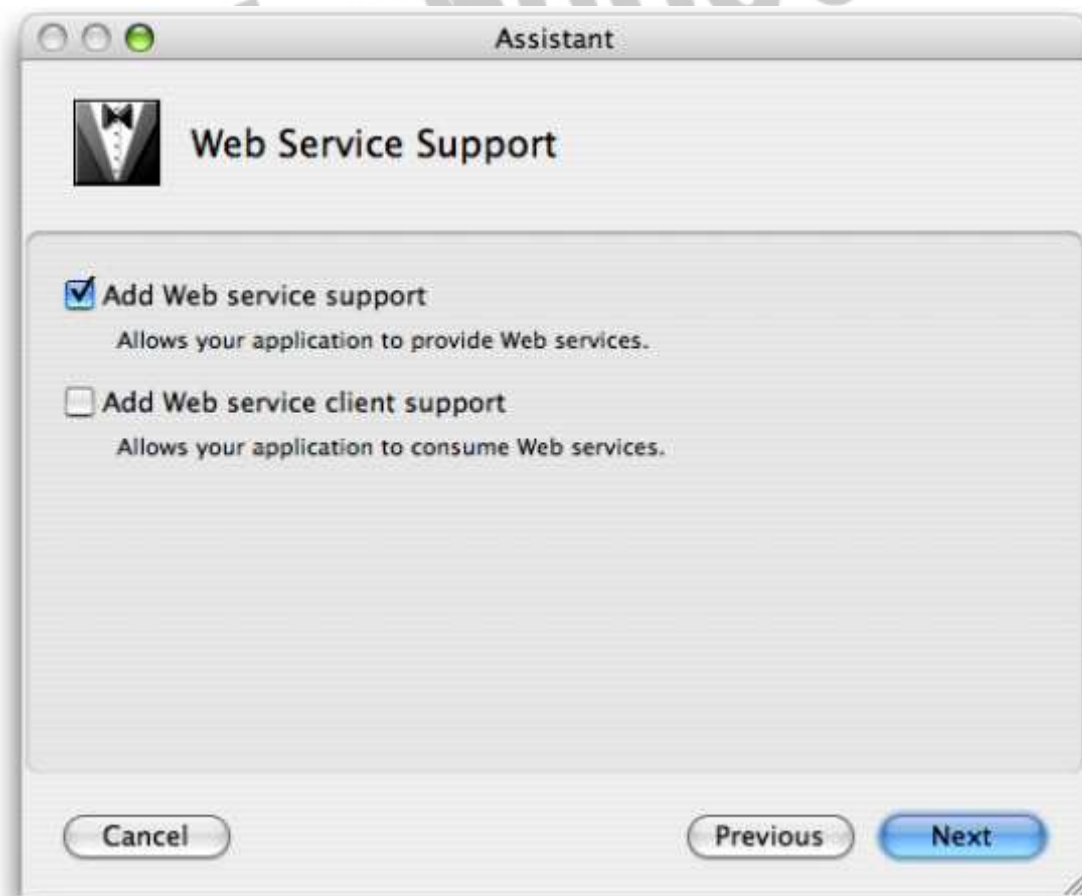


شکل ۴-۵: وارد کردن نام پروژه



۵- اگر می خواهید برنامه وب خود را در کانتینر سرولت J2EE آرایش دهید، گزینه “Deploy a servlet container” را انتخاب کرده و سپس Next را کلیک کنید.

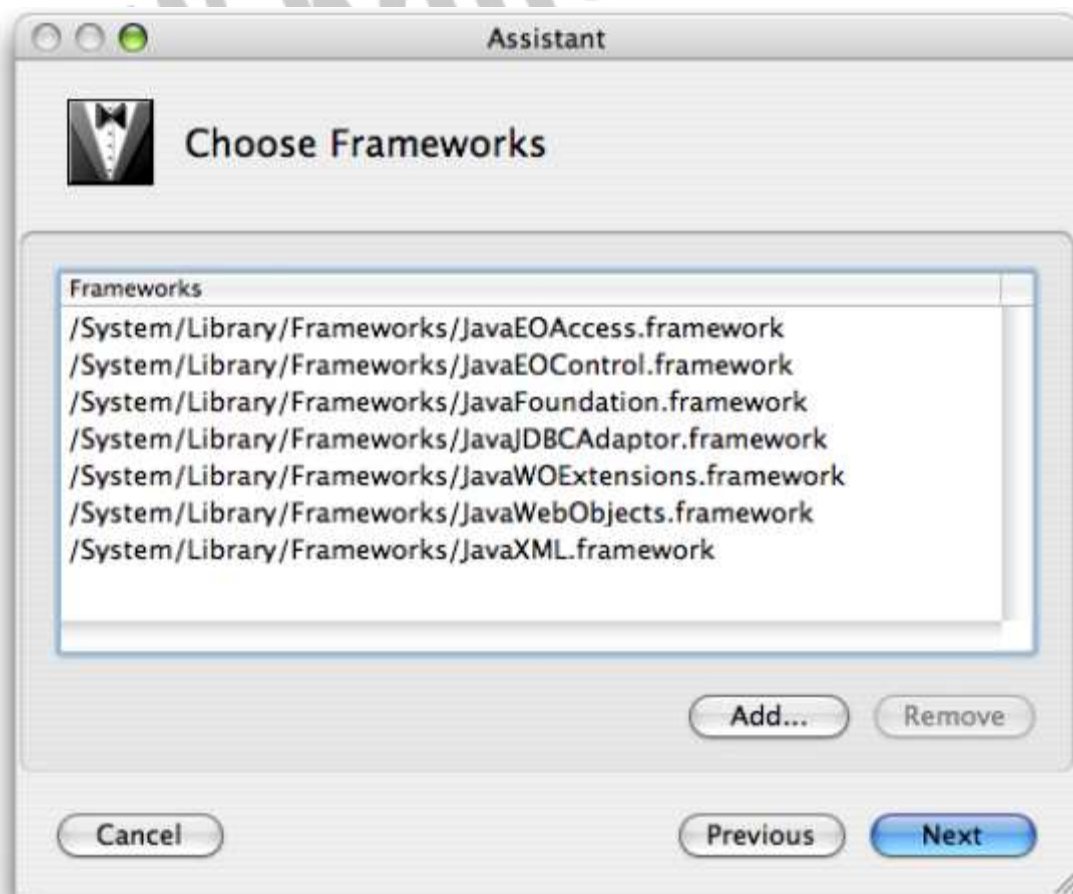
۶- اگر برنامه شما یک سرویس وب است، گزینه “Add web service support” را در کادر web service support انتخاب کنید. اگر برنامه شما از سرویس وب استفاده می کند، بر روی “Add web service client support” کلیک کنید. سپس نظیر شکل ۳ Next را کلیک کنید. استفاده از سرویس های وب اختیاری است.



شکل ۴-۶: افزودن پشتیبانی سرویس وب

۷- اگر از آداپتور JDBC استفاده می کنید، در کادر Choose EOAdaptor گزینه Java JDBCAdaptor. Framework را انتخاب کنید. اگر از آداپتور JNDI استفاده می کنید، Java JNDIAdaptor. Framework را انتخاب کنید.

آداپتور پیش فرض دیتابیس JDBC است زیرا اکثریت دیتابیس های مدرن از JDBC پشتیبانی می کنند.  
اگر مطمئن نیستید از چه دیتابیزی استفاده می کنید، فقط Next را کلیک کنید.  
۸- اگر نیاز است چارچوب های اضافی را به پروژه خود- به عنوان مثال، چارچوب های دیتابیس دسته سوم- اضافه کنید، نظیر شکل ۴ در کادر Choose Framework روی Add کلیک نمایید. در غیر این صورت، برای ادامه Next را کلیک کنید.  
Assistant چارچوب های مناسب webobject و جاوا را به پروژه شما اضافه می کند که به الگوی انتخابی شما بستگی دارد. اگر افزودن چارچوب های بیشتری ضروری نیست، صرفاً Next را کلیک کنید.



شکل ۴-۷: انتخاب چارچوب ها

۹- سپس با کلیک Add در کادر EOModels، یک مدل EO را اضافه کنید.  
اگر مدل EO موجودی دارید که یا با استفاده از Xcode یا Eomodeler خلق کردید، حالا می توانید آن را به پروژه خود اضافه کنید. اگر الگوی Direct to web Application یا Display Group Application انتخاب کردید، سپس انتخاب مدل EO الزامی است.

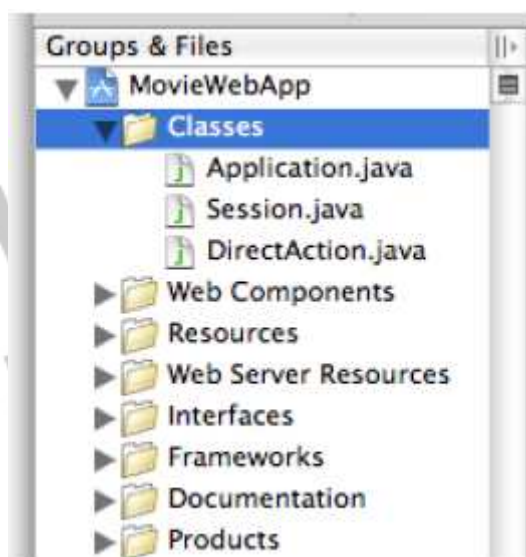
- ۱۰- اگر الگوی web Application را انتخاب کردید، Finish را کلیک کرده و به مراحل بعدی بروید.
  - ۱۱- اگر الگوی Direct to web Application را انتخاب کردید، سپس کادرهای محدود خاصی در webobjects Direct to web Guide, برای چگونگی خلق یک برنامه، Direct to web را مطالعه کنید.
  - ۱۲- اگر الگوی Direct to web Application را انتخاب کردید، بعداً کادرهای محدود خاص پیکربندی یک گروه نمایش پیدا می شود.
  - ۱۳- مدخل اصلی در کادر Choose the main EOEntity را انتخاب کنید. مدخلی را انتخاب کنید که شیء های ریشه را نشان می دهد و Next را کلیک کنید.
  - ۱۴- در کادر Choose a layout طرح اولیه ای برای صفحه انتخاب کنید.
  - ۱۵- ویژگی هایی برای نمایش صفحه مشابه پیکربندی برنامه Direct to web را در کادر Choose Attributes to display را انتخاب کنید. سپس Finish را کلیک کنید.
- هنگامی که دکمه Finish را کلیک کردید، پنل Assistant بسته شده و پنجره شامل همه فایل های برنامه شما باز می شود. در مورد تشریح فایل های پروژه شما، قسمت "فایل ها و گروه های پروژه" را مطالعه کنید.

#### ۸-۴-۳ فایل ها و گروه های پروژه

فایل ها و گروه های نمایش در Xcode متفاوت هستند که به نوع الگوی webobject انتخابی شما هنگام خلق پروژه Xcode بستگی دارد. این قسمت برخی از گروه هایی را تشریح می کند که هنگام خلق یک برنامه وب آشکار می شود.

#### ۸-۴-۴ کلاس ها

همانطور که در شکل ۵ نشان داده می شود گروه Clases از کلاس های جاوایی تشکیل می گردد که با مؤلفه های وب تناظری ندارند. به طور نمونه، این گروه Direct Action. Java, Session. Java Application. Java را در برمی گیرد. کلاس های متناظر با مؤلفه های وب در گروه web Components قرار دارند. به عنوان مثال، Main. Java در Main / web Components قرار دارد.



#### شکل ۴-۸ گروه کلاس ها

کلاس های پیش فرض در پروژه webobject عبارتند از:

- Application. Java زیر مجموعه ای از WoApplication است. هنگامی شیء برنامه به طور اتوماتیک خلق می شود که برنامه شما آغاز شده و با نمونه برنامه متناظر است.
- Session. Java زیر مجموعه ای از Wo Session است. هنگامی شیء Session به طور اتوماتیک خلق می شود که کاربر ارتباطی با برنامه وب شما برقرار کند.
- Direct Action. Java زیر مجموعه ای از Wo Direct Action است.

#### ۴-۹ مؤلفه های وب

همان طور که در شکل ۶ نشان داده می شود، گروه مؤلفه های وب همه فایل هایی را در بر می گیرد که به مؤلفه های وب متعلق است. یک مؤلفه وب، یک صفحه یا بخشی از یک صفحه را در برنامه شما نشان می دهد. یک برنامه می تواند از یک یا چندین مؤلفه وب برخوردار باشد.

به عنوان مثال، هر برنامه webobject حداقل مؤلفه ای به نام Main دارد که در گروه Webcomponents وجود دارد. یک فولدر مؤلفه وب از چند فایلی تشکیل می شود که رفتار و ظاهر مؤلفه را مشخص می



کند. هر فایلی از پیشوند مشابه اما با پسوندهای مختلفی برخوردار است. اینها فایل های موجود در یک فولدر مؤلفه وب هستند:

- مؤلفه وبی با پسوند wo. که طرح اولیه عناصر HTML و اتصال ها را در عناصر پویا ذخیره می کند.  
- فایل کلاسی با پسوند java. که رفتار مؤلفه را پیاده سازی می کند. هر یک از مؤلفه ها زیر مجموعه ای از Woxomponent است. به طور نمونه، کلاس شما متغیرها و متدهای محدود به عناصر پویا را پیاده سازی می کند.

- یک فایل API با پسوند api. که کلیدهای تعریف شده توسط مؤلفه ای که مؤلفه های دیگر به آن متصل شده و فرمان هایی برای اتصال کلیدها را دربرمی گیرد. webobjects Builder از این فایل ها استفاده کرده تا چک کند که آیا مؤلفه قابل استفاده مجدد به درستی به کار رفته است یا خیر.  
به طور نمونه مؤلفه ای را با استفاده از webobjects Builder ویرایش کنید- فقط روی فولدری با پسوند wo. دوبله کلیک کنید تا آن را در webobjects Builder ویرایش نمایید. به هر حال، اگر فرمت را درک کنید، گاهی می توانید این فایل ها را مستقیماً ویرایش کنید. به عنوان مثال، این ها فایل های فولدر Main.wo هستند:

- main. Html الگوی HTML برای مؤلفه است. این فایل درست مثل هر وب پیجی از تگ های HTML برخوردار است. به علاوه می تواند تگ های عناصر پویا را دربرداشته باشد.

- main.wod فایل اعلام هایی است که ارتباطات بین عناصر پویا و متغیرها یا متدهای را در فایل جاوا شما مشخص می کند.

- mian.woo برای ذخیره سازی اطلاعات در مورد گروه های نمایشی- مثلاً، اگر پروژه شما به دیتابسی دسترسی داشته باشد- و رمز گذاری برای الگوهای HTML به کار می رود. به ندرت این فایل را مستقیم ویرایش کنید.

#### ۴-۱۰ منابع (Resources)

گروه Resources فایل هایی را در بر می گیرد که برنامه شما در زمان اجرا به آن نیاز دارد اما وجود آن در ریشه پرونده سرور وب ضروری نیست و از این رو، کاربران به آن دسترسی نخواهند داشت. فایل های Resources می تواند فایل های متفرقه پیکربندی، فایل های مدل EO و آیکون ها را در بر داشته باشد.

#### ۴-۱۰-۱ منابع سرور وب

گروه web server Resources فایل هایی نظیر تصاویر و صوت را در بر می گیرد که باید تحت ریشه document سرور وب در زمان اجرا باشد. هنگام توسعه برنامه، این فایل ها را در دایرکتوری پروژه خود قرار داده و آنها را به گروه web server Resources پروژه اضافه کنید. هنگامی که پروژه خود را

می سازید، Xcode فایل های این گروه را در فولدر web server Resources پوشه برنامه خود کپی می کند.

#### ۴-۱۰-۲ چارچوب ها

گروه Frame works چارچوب های انتخابی شما در Assistant حین ایجاد پروژه webobject و همینطور چارچوب هایی را در بر می گیرد xcode به طور اتوماتیک به پروژه شما اضافه می کند. چارچوب مجموعه ای از کلاس ها و منابعی است که برنامه های چندگانه به کار می برند. با ذخیره سازی آیتم هایی چون مؤلفه ها و تصاویر در چارچوب ها، می توانید آنها را بدون خلق کپی های چندگانه در پروژه های چندگانه مجدداً استفاده کنید.

هر پروژه webobjects به طور پیش فرض از چندین چارچوب برخوردار است که به الگوی انتخابی شما در Xcode بستگی دارد. یک برنامه webobjects که از Enterprise (داده های سازمانی) استفاده می کند این عناصر را در بر می گیرد:

– Java EOControl- framework با لایه کنترل در Enterprise objects مطابقت دارد که زیرساختاری برای ایجاد و مدیریت شیء های سازمانی را فراهم می آورد.

– Java EOAccess- framework با لایه Access (دسترسی) در Enterprise objects مطابقت دارد که مکانیزم های دسترسی دیتا را برای تکنولوژی Enterprise objects فراهم می کند.

برای کسب اطلاعات بیشتر در مورد داده های سازمانی، webobjects Enterprise objects, objects programming guide (راهنمای برنامه نویسی داده های سازمانی webobjects) را مطالعه نمایید. اگر آداپتور JDBC را برای دسترسی به دیتابیس خود انتخاب کردید، پروژه شما از این چارچوب تشکیل می شود:

– Java JDBCAdaptor. Framework پیاده سازی یک آداپتور داده های سازمانی برای منابع دیتای JDBC را ارائه می کند.

یک برنامه وب این چارچوب ها را در بر می گیرد:

– Java foundation. Framework مجموعه ای از کلاس های کامل و نیرومند هسته را فراهم می کند که شامل کلاس های برنامه سودمند، جمع آوری، کدگذاری مقدار کلید، زمان و تاریخ، و ثبت وقایع اشکال زدایی می شود.

– Java webobject. Framework کلاس های سرور هسته برنامه وب، مدیریت session، مؤلفه وب و حلقه درخواست- پاسخ را در بر می گیرد.

– Java WOExtensions. Framework مؤلفه های اضافی قابل استفاده مجدد وب را در بردارد.

– JavaXML. Framework پشتیبانی محتوای XML- به عنوان مثال، تجزیه کننده Apache XML، را در بردارد.

یک برنامه Direct to web این چارچوب ها را در بر می گیرد.

– Java DTGeneration. Framework پشتیبانی ایجاد صفحات Direct to web را فراهم می کند.

– Java Direct to web. Framework کلاس هایی برای توسعه سریع برنامه های وب مبتنی بر HTML را فراهم می آورد.

– Java EOProjects. Framework سرویس هایی برای webobjects Builder ارائه می دهد- به عنوان مثال، کلیدها و اقدامات را از یک فایل جاوا مؤلفه وب برگشت می دهد.

#### ۴-۱۰-۳ محصول (products)

بعد از آنکه برنامه خود را ساختید، گروه products شامل wrapper برنامه می شود که فولدري به اسم نام پروژه با پسوند woa می باشد- به عنوان مثال، اگر نام پروژه My wb App باشد پس، wy web App. Woa پوشه (wrapper) برنامه ساختاری مشابه ساختار چارچوب دارد که موارد ذیل را در بر می گیرد:

– برنامه قابل اجرا- به عنوان مثال wy web App.

– منبع برنامه در گروه Contents.

گروه Resowrec از مؤلفه های وب برنامه و همین طور فایل های دیگری تشکیل می شود که برنامه شما در زمان اجرا به آن ها نیاز دارد.

– اطلاعات سرور وب برنامه کاربردی در گروه Contents. اگر برنامه شما فاقد اطلاعات سرور وب باشد، پس گروه web server Resurces در گروه Contents پدیدار نمی شود.

هنگامی که برنامه خود را ساخته و نصب کردید، Xcode همه فایل های گروه web server Resurces شما را در فولدري به نام web server Resurces در پوشه برنامه کپی می کند. اگر مؤلفه های جاوا Client- slide در پروژه خود دارید، آنها هم در فولدر web server Resurces کپی می شود.

#### ۴-۱۱ مقصد ها

مقصدهای یک برنامه کاربردی وب عبارتند از:

– برنامه کاربردی- به عنوان مثال My web App.

– Application Server قسمتی از برنامه شما را می سازد که مؤلفه های وب و شیءهای سازمانی را خلق می کند.

– web server منابعی می سازد که می تواند توسط سرور HTTP به کار رود مثل تصاویر و فیلم های Quik time ذخیره نشده در دیتابیس.

## ۴-۱۲ ساخت برنامه خود

ساخت و نصب برنامه کاربردی وب شما ساده است. فقط مقصد برنامه را انتخاب کرده و دکمه Build and go در X Code را کلیک کنید. برای اجرا و تست برنامه خود از مرورگر وب استفاده کنید. به عنوان مثال، اگر الگوی Direct to web را انتخاب کردید، برنامه Direct to web شما ساخته و آغاز می شود. Safari همچنین آغاز شده و با URL برنامه Web objects به برنامه شما متصل می شود.

## ۴-۱۳ نصب برنامه کاربردی خود

ممکن است بخواهید برنامه را برای تست در ماشین توسعه خود نصب کنید. قبل از نصب یا آرایش برنامه، باید چگونگی کار سرور وب و مکان مورد نیاز نصب فایل ها را درک کنید. برخی فایل ها در برنامه وب - مثل فایل صوتی و تصویری - باید تحت ریشه document سرور وب ذخیره شود تا سرور به آنها دسترسی داشته باشد. زیرا فایل ها بخشی از HTML پویا بوده که سرور وب به کلانیت های وب ارسال می کند. فایل های باقی مانده - نظیر کد سرس و مؤلفه های شما - باید برای برنامه شما و نه الزاماً خود سرور وب قابل دسترسی باشد. از این رو، هنگامی که یک برنامه وب را نصب کرده یا آرایش می دهند، فایل های محصول شما تقسیم می شود - این فایل های مورد نیاز سرور وب در ریشه document قرار داده شده و همه فایل های دیگر در جای دیگری ذخیره می شود. به این نوع نصب، نصب دو بخشی (Split install) گفته می شود.

## ۴-۱۴ خلق مؤلفه های وب

مؤلفه های وب در چگونگی کار محتوای پویا در webobject اساسی هستند. بطور نمونه، اگر اطلاعات روی وب سایت شما غالباً تغییر بیابد یا براساس برخی شرایط تغییر می کند Nebobjects را انتخاب کنید. مثالهایی از وب سایتهای پویا عبارتند از خبرهای آنلاین، فروشگاهها، رأی گیری و آمار و ارقام. Webobjects هم برای هر وب سیتی ایده آل است که جلسات کاربر را پیگیری کرده و سرویس های شخصی نظیر تألیف محتوا و صفحات رایج ارائه می دهد. محتوای پویا را می توان برای اولویت های کاربر و معیار جستجو تنظیم کرد.

مؤلفه اهی وب را در نمایش صفحات وب یا صفحات وب جزیی بوجود آورده از سوی وب سایت شما استفاده کنید. مؤلفه های وب در واقع الگوهایی برای ایجاد صفحات HTML هستند. مؤلفه های وب از عناصر پویا و ایستا ساخته می شود. برای متصل کردن نقاط مقابل HTML به متغیرها و متدهای کلاس مؤلفه وب خود از عناصر پویا استفاده کنید. برخی عناصر انتزاعی بوده و صرفاً در کنترل ایجاد HTML - مثلاً شرایط و تکرارها - بکار می رود.



بطور نمونه، اگر چه می توانید برنامه وبی را بدون استفاده از Enter prise Objects (شیء های سازمانی) خلق کنید اما، وب سایت شما صفحات HTML را ارائه می دهد که با دیتای بدست آمده از شی های سازمانی ذخیره شده شما در یک دیتابیس Back-end مشهور شده اند. از این رو، مؤلفه های وب با میانجی کردن بین ویوها (عناصر پویا) و مدل های شما (شیء های سازمانی) شبیه کنترلرهای الگوی طراحی MVC رفتار می کنند. همچنین می توانند از عناصر انتزاعی استفاده کرده و گروهها - کنترلرهایی که شیء های سازمانی زیادی را دستکاری می کند را به روش های جالب خلق صفحات وب هوشمند نشان می دهد.

مؤلفه های وب از تمامی مزایای سیستم های شیء گرا بهره می گیرد. مؤلفه های وب قابل استفاده مجدد، حفظ و توسعه است. مؤلفه های وب می تواند مؤلفه های دیگر وب - مؤلفه هایی که صفحات کم اهمیت را نشان می دهند - همینطور عناصر پویا، عناصر ایستا و تکست واضح را در بر داشته باشد. هر تگ HTML را می توان به الگوی HTML یک مؤلفه وب اضافه کرد.

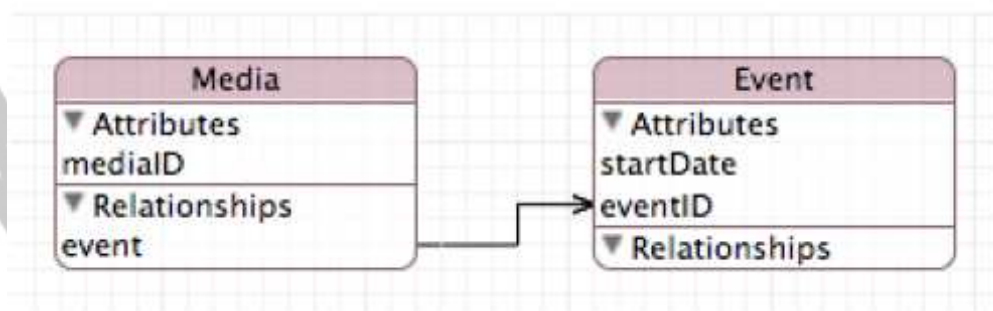
مؤلفه های وب فولدرهایی هستند که به پروژه X Code خود اضافه می کنید. هر فولدری از یک فایل HTML، WOD، API و Java تشکیل می شود. فایل HTML قالب را نشان می دهد، فایل WOD اتصالات پویای عنصر را در بر می گیرد، فایل API هر نوع اتصالاتی را در بردارد که مؤلفه شما صادر می شود و فایل Java کلاس است که منطق کنترلر را پیاده سازی می کند. منطق کنترلر خود - شیوه ها و متغیرها - را با استفاده از X Code به کلاس Java اضافه کنید. می توانید فایل های دیگر را مستقیماً ویرایش کنید. اما به طور نمونه، از Web Objects Builder - یک ویرایشگر گرافیکی - در طراحی مؤلفه های وب خود استفاده کنید. Web Objects Builder فایل های واقع در فولدر مؤلفه وب را خلق می کند.

این قسمت چگونگی استفاده مجدد و توسعه مؤلفه های وب را از دیدگاه یک برنامه نویس زبان جاوا تشریح می کند. برای اطلاعات بیشتر پیرامون چگونگی خلق گرافیکی مؤلفه های وب، Web Object Builder user Guide را مطالعه کنید.

اگر از دیتابیس back - end در ذخیره سازی شیء های سازمانی خود استفاده می کنید، باید ابتدا مدل EO خود را خلق کرده و سپس پروژه X Code خود را قبل از خلق مؤلفه هی وب ایجاد کنید. برای خلق شیء های معمولی سازمانی و مدل EO قسمت "ایجاد شیء های سازمانی" را مطالعه کنید. در مورد چگونگی ایجاد یک پروژه X Code قسمت "ایجاد پروژه ها" را مطالعه نمائید.

بطور پیش فرض، هر برنامه Web Objects یک مؤلفه Mian در بردارد. این مؤلفه اساساً؟؟ اولین صفحه نمایش داده شده به کاربران است مگر آنکه شما مشخص کنید آن صفحه Login یا تونل بقیه برنامه شماست.

به عنوان مثال، اگر هنگام خلق پروژه Xcode خود الگوی Web Application را انتخاب می کنید، همچنان که در شکل ۱ نشان داده می شود، Main.Wo در گروه Web Components آشکار مشخص می شود. برای باز کردن مؤلفه Main در Web Objects Builder، روی Main.wo را دوبله کلیک کنید.



شکل ۴-۹: فایل های مؤلفه وب

## ۴-۱۵ فایل های Java

هر مؤلفه وب از یک فایل Java برخوردار است که آنرا در اضافه کردن منطق کاری و کنترلر استفاده می کنید.

به عنوان مثال، اگر مؤلفه Main شما یک صفحه Login باشد، می تواند یک فیلد 'username' و پسورد در برداشته باشد تا به کاربر اجازه دهد به وب سایت شما وارد شود، سپس ممکن است متغیرهای 'username' و 'Pass word' داشته باشید تا متن وارده کاربر را ذخیره کند. بعلاوه، زمانی که کاربر Return را وارد کرده یا دکمه Login را کلیک می کند، به متد عمل (action) نیاز دارید. مانند شکل یک، این متغیرها و متدها را به Main.Java موجود در گروه Web Components اضافه کنید. بعد، عناصر پویا، عناصر Wotextfield، WopasswordField و NosubmitButton را با استفاده از Web Objects Builder به متدها و متغیرهای خود اضافه کنید.

بطور متناوب می توانید با استفاده از Web Objects Builder که فایل Java را ویرایش می کند، متدها و متغیرها را به مؤلفه خود اضافه نمایید.

#### ۴-۱۵-۱ فایل های WOD و HTML

گاهی ممکن است به ویرایش مستقیم یکی از فایل های مؤلفه وب نیاز داشته باشید. به هر حال، از آنجا که این فایلها به وسیله Web Objects Builder بوجود می آید، باید قبل از ویرایش، از فرمت این فایلها آگاه باشید. اگر فرمت این فایل ها را تغییر می دهید، ممکن است مؤلفه متغیر نشود یا در Web Objects Builder باز نشود.

به عنوان مثال، فرض کنید یک مؤلفه Main را با استفاده از Web Objects Builder ایجاد می کنید که یک رشته تکست را در بردارد، "The current time is" به وسیله عنصر Wostring دنبال می شود که زمان فعلی را نشان می دهد. بعلاوه، عنصر Wostring را به متد Current Time مؤلفه Main خود متصل کنید که زمان فعلی را برگشت می دهد. این اطلاعات در فایل های WOD و HTML مؤلفه وب ذخیره شده که شما آنها را در X Code می بینید.

فایل HTML موجود در لیست یک از عنصر WEBOBJECT نشانگر موقعیت برخوردار است که wostring مقدار برگشتی توسط متد Current Time را تعیین می کند. توجه کنید که عنصر به شکل `<WEBOBJECT NAME=String 1 ></WEBOBJECT>` تعریف می شود. یک مدخل متناظر در فایل WOD وجود داشته که از همان نام، string، استفاده می کند.

لیست یک - نمونه فایل HTML

ارتباط بین عنصر Wostring و متد Current Time در فایل Wod لیست ۲ اعلام می شود. مدخل صرفاً یک اتصال لیست شده، ارتباط بین صفت Value و متد Current Time دارد. زمانی این متد فراخوانی می شود که عنصر Wostring نیاز به نمایش مقدار خود دارد. فایل WOD تنظیمات عنصر دیگری را در بر می گیرد که در Webobjects Builder آشکار نمی شود.

لیست فایل WOD :

چگونگی عملکرد عناصر پویا:

هنگام برنامه نویسی با مؤلفه های وب، به درک چگونگی عملکرد عناصر پویا در زمینه حلقه درخواست پاسخ کمک می کند.

وقتی یک برنامه ساده ای را اجرا می کنید که زمان فعلی را در صفحه اصلی نظیر آنچه در قسمت "فایل های WOD و HTML" شرح داده شد، نشان می دهد، صفحه نمایشی مرورگر وب عنصر Wostring را جایگزین می کند که با زمان فعلی به مؤلفه Main اضافه کردید. اگر صفحه را دوباره لود کردید، تایم آپدیت می شود. WebObjects صفحه را در طی حلقه درخواست - پاسخ بطور پویا اسمبلی می کند.

وقتی به URL متناظر با برنامه خودتان در مرورگر وب دسترسی دارید، سرور وب آداپتور WebObjects را کنترل می کند - فرآیندی که نمونه های برنامه WebObjects را به سرورهای وب مرتبط می سازد. این برنامه از طریق صدها مرحله پاسخ را ایجاد می کند.

۱- فایل HTML را مطالعه کنید.

WebObjects مانند یک سرور وب معمولی ابتدا فایل HTML را مطالعه می کنند.

۲- عناصر WebObjects را رندر کنید.

WebObjects برخلاف یک سرور وب معمولی عناصر WebObject را قبل از بکارگیری فایل در سرور وب تجزیه می کند.

هنگامی که WebObjects با عنصر WEBOBJECT مواجه می شود، فایل WOD را برای مؤلفه متناظر وب می سازد. همه عناصر WEBOBJECT در فایل HTML قالب مؤلفه وب نام گذاری شده و نظیر لیست ۲، هر یک بوسیله نام خود در فایل WOD فهرست می شود.

هر نوعی از عنصر WebObjects از منطق خاصی برای ساخت کد HTML برخوردار بوده تا به سرور وب برگردد. متعارف سازی این فرآیند با صفات تعریف شده توسط برنامه نویس مؤلفه وب صورت می گیرد. هر اتصالی در یک فایل WOD یا ایستا است یا پویا. اگر اتصالی ایستا باشد، مقدار فراهم شده مستقیماً استفاده می شود.

اگر اتصالی پویا باشد - یعنی صفتی به متغیر نمونه یا متد مرتبط باشد WebObjects متد را استنتاج کرده یا برای بدست آوردن مقدار در زمان اجرا به متغیر نمونه دسترسی دارد. به عنوان مثال، هنگامی که عنصر Wostring ارزیابی می شود، متد نام گذاری شده در اتصال Value خود CurrentTime را استنتاج می کند تا مقدار نمایش را بدست آورد. پیاده سازی Wostring شیء Ntimestamp را به تکست تبدیل می کند که در صفحه وب برگشتی به مرورگر وب بوجود می آید. قبل از برگشت Webpage به مرورگر، Webobjects کد HTML مؤلفه نشان داده شده در لیست ۳ را به markup مشابه markup لیست ۴ تبدیل می کند.

لیست ۳- تفسیر کد HTML توسط WebObjects

لیست ۴- ارسال WebObjects کد HTML به مرورگر وب

این فرآیند هر باری روی می دهد که مرورگر وب صفحه Main را درخواست می کند. اگر دوباره صفحه را لود کنید، متد دوباره استنتاج شده و مقدار تایم جدیدی نمایش داده می شود. برای کسب اطلاعات بیشتر در مورد حلقه درخواست - پاسخ، قسمت "حلقه درخواست - پاسخ" را مطالعه نمایید.



## ۴-۱۵-۲ حفظ و نگهداری وضعیت

درک رابطه بین فایل های Java ، WOD و HTML مؤلفه وب بخش مهمی از توسعه WebObjects به شمار می آید. نه تنها متدها و متغیره را به مؤلفه خود اضافه کرده تا عناصر پویا را مقید سازد، بلکه همچنین می توانید بر متدها و متغیرها اضافه کرده تا وضعیت یک مؤلفه را حفظ کند.

هنگامی که متدها را به مؤلفه ای در WebObjects Builder اضافه می کنید، در واقع فایل سرس Java مؤلفه را ویرایش می کنید. وقتی چگونگی ظاهر مؤلفه را با افزودن عناصر اصلاح می کنید، خود فایل HTML را اصلاح می نمائید، وقتی عناصر پویا را به متغیرهای مؤلفه خود متصل می نمائید، خود فایل WOD را اصلاح می کنید.

بطور نمونه، بعد از استفاده WebObjects Builder در تعریف اجزاء اصلی یک مؤلفه وب می توانید جزئیات را با ویرایش مستقیم فایل های Java و HTML اضافه کنید (به ندرت به ویرایش یک فایل WOD نیاز می باشد). برای ویرایش فایل HTML یک مؤلفه وب در WebObjects Builder، به آسانی از مد Source استفاده کنید - دکمه Source در منو ابزار را کلیک کرده تا از ویرایشگر HTML استفاده کنید. می توانید فایل جاوا را با استفاده از X Code ویرایش نمائید.

هنگامی که برنامه خود را آرایش داده و کاربران از طریق مرورگر وب به وب سایت شما وصل می شوند، صفحات وب طبق نیاز از مؤلفه های وب ایجاد می شود. در واژگان برنامه نویسی، مؤلفه های وب همگی زیر مجموعه های Wocomponent بوده و طبق نیاز معرفی می شود. به عنوان مثال، وقتی کاربر مؤلفه main مشروح در قسمت "فایل های WOD و HTML" را درخواست می کند، نمونه main خلق می شود. وقتی که زمان اضافه کردن WebObjects به محتوای Wostring می باشد، Value عنصری را جستجو می کند که به فایل WOD متصل است. اتصال Value با متد Current Time تنظیم می شود. از این رو، WebObjects، متد Current Time را به مثال مؤلفه وب می فرستد که زمان فعلی را برگشت می دهد.

مثالی از یک مؤلفه وب حداقل در دوچرخه حیات حلقه درخواست - پاسخ زنده است. در چرخه اول، صفحه وب رندر و ارائه می شود و در چرخه دوم، مؤلفه مشخص می کند که بعداً کدام صفحه نشان داده می شود. اگر صفحه مشابه صفحه قبل نباشد، WebObjects مثالی از مؤلفه جدید را خلق می کند. سپس، مؤلفه قدیمی حذف شده یا در حافظه کش سرور ذخیره می شود تا به کاربران اجازه دهد تا پیمایش معکوس به صفحات قبلی داشته باشند. بهر حال، اگر مؤلفه نمایش مشابه و یکسان است، مثال به حیات خود ادامه می دهد. در این مورد، ورژن قبلی مؤلفه در حافظه کش Back track ذخیره می شود.

جهت کسب اطلاعات بیشتر پیرامون عمل یک تراک (پیمایش معکوس) قسمت "مدیریت حفاظه کش و بک تراک" را مطالعه کنید.

روش دیگر استفاده از متغیرها و متدها در فایل سرس Java، حفظ وضعیت دیگر می باشد مثلاً، پیگیری اقدامات کاربر هنگامی که با برنامه شما رابطه متقابل دارد.

مثال: نمایش شمارش صفحه

قسمت های زیر چگونگی حفظ وضعیت در مؤلفه وب شما را با پیاده سازی یک برنامه ساده وب نشان می دهد که تعداد دفعات نمایش مؤلفه main توسط یک کاربر واحد را می شمرد. این مثال کنتورو هایپرلینک Refresh صفحه را به مؤلفه main اضافه می کند. برای حفظ وضعیت کنتور برای دوره جلسه از مؤلفه Web استفاده می کند.

ایجاد پروژه شما

یک برنامه ساده وب را بدینگونه خلق کنید:

۱- X Code را شروع کرده و منوی File > New Project را انتخاب کنید.

۲- الگوی برنامه WebObjects را انتخاب کرده و Next را کلیک کنید.

۳- در کادرهای محاوره ای مانده روی Next کلیک کرده تا همه مقادیر پیش فرض را انتخاب کنید.

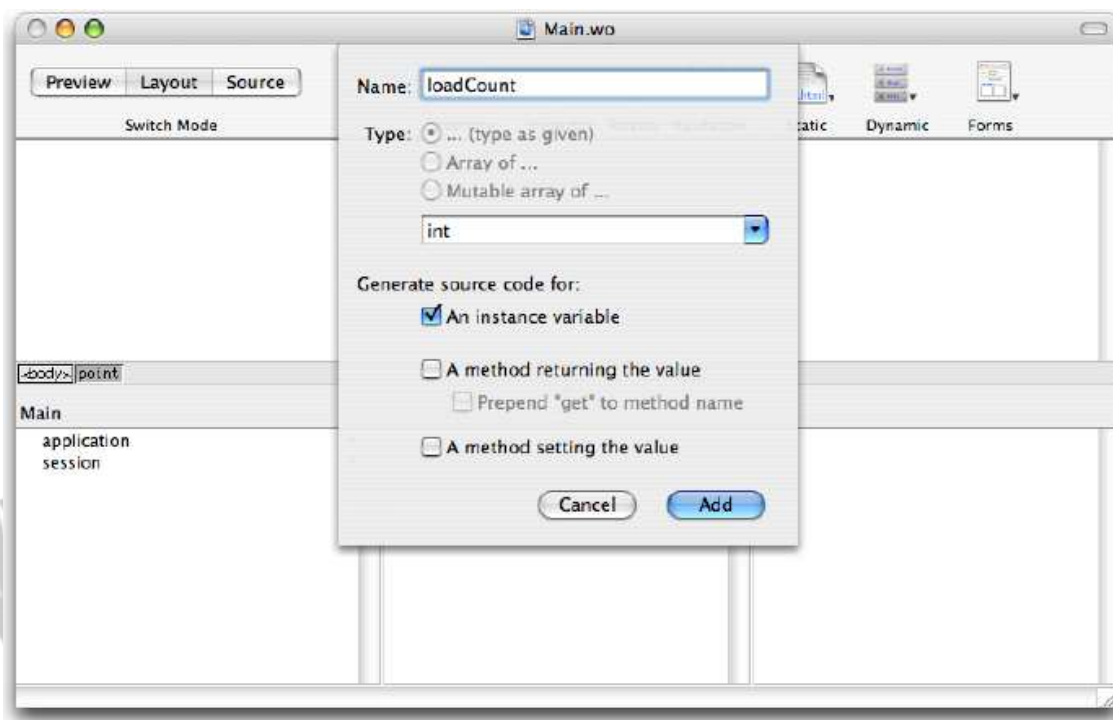
۴- روی Finish در کادر نهایی کلیک کرده تا پروژه را بوجود آورید.

افزودن متغیرها به مؤلفه خود

ابتدا متغیر و کدی را به مؤلفه Main اضافه کنید تا با شمارش با هر بار نمایش صفحه افزایش یابد.

۱- Main.Wo در Web Objects Builder را با دوبله کلیک آن در X Code باز کنید.

۲- Add Key را از منو Inter face در منو ابزار پنجره Main.wo انتخاب کنید.



شکل ۴-۱۰: اضافه کردن کلید

۴- فایل Main.java را بررسی کرده تا تأیید کند که متغیر اضافه شده است. نظیر لیست کد را برای شروع Load Count اصلاح کنید.

لیست ۵- افزودن متغیر

```
public class Main extends W0Component {  
    public int loadCount = 1;  
  
    public Main(W0Context context) {  
        super(context);  
    }  
}
```

افزودن عناصر پویا به مؤلفه خود

سپس از متغیر در نمایش تعداد دفعات لود صفحه استفاده کنید. برای نمایش شمارش لود در صفحه وب، نیاز بوده عنصر Wostring را با استفاده از WebObjects Builder به مؤلفه Main اضافه کنید.

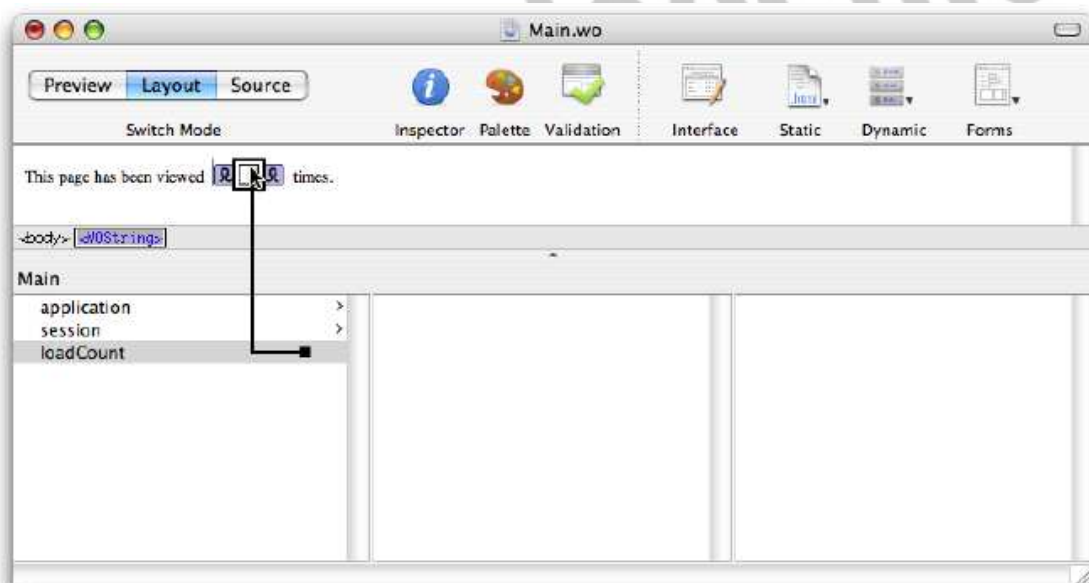
۱- لیل و عصر Wostring را به Main.Wo اضافه کنید.

الف – This page has been viewed را وارد کنید.

ب – فاصله و عنصر Wostring را به سمت راست لیبِل اضافه کنید.

ج – فاصله و times را به سمت راست عنصر Wostring اضافه کنید.

۲- مانند شکل ۳، از Loadcount را در مرورگر شیء به عنصر Wostring درآگ کنید تا آنرا به صفت Value عنصر Wostring متصل کنید.



شکل ۴-۱۱: اتصال Wostring

## ۴-۱۶ افزودن متدها به مؤلفه

حالا، افزودن یک روش ضروری بوده تا صفحه را با استفاده از hyperlink دوباره لود کنید. عناصر WoHyperlink و هایپرلینک های معمولی در WebObjects متدها را فراخوانی می کنند - متدها در فایل سرس Java مؤلفه تعریف می شود. این متدها، شیوه های action (اقدام) خوانده می شود. همه متدهای اقدام یک مؤلفه وب نمایشگر صفحه بعدی را برگشت می دهد. اگر یک متد اقدام null را برگشت دهد، پس دوباره همان صفحه نمایش داده می شود. جزئیات بیشتر متدهای action در قسمت پردازش درخواست پوشش داده می شود.

مراحل زیر را برای افزودن یک متد اقدام refresh time دنبال کنید.

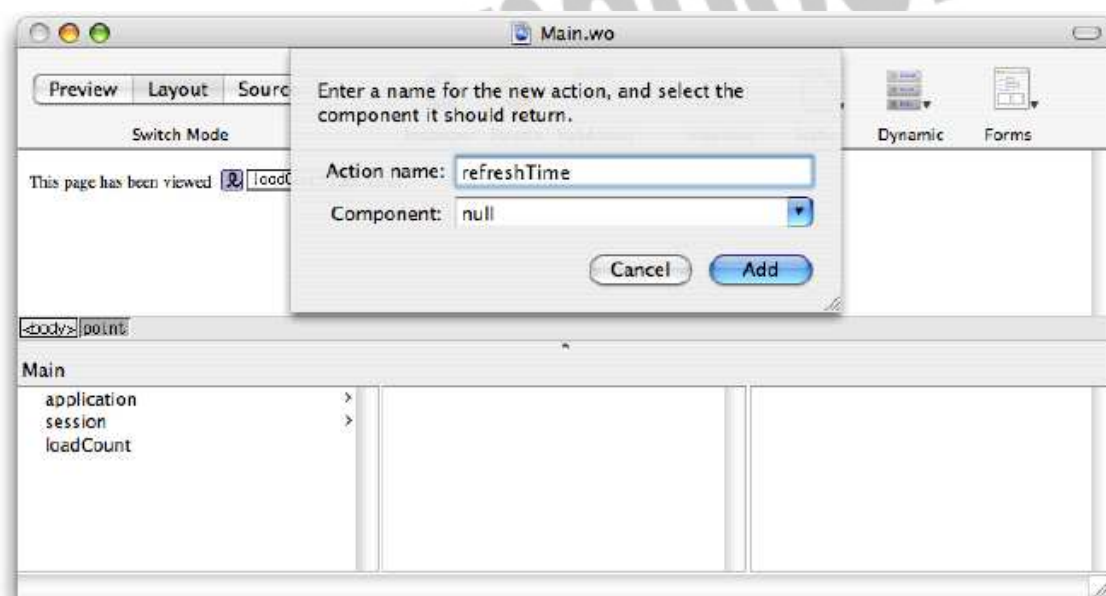
۱- متد اقدام را اضافه کنید.



الگوی مؤلفه Main در WebObjects Builder را باز کرده Add Action را از منو Interface منو ابزار انتخاب کنید.

الف - عمل refresh time را نام گذاری کنید.

ب - همچنانکه در شکل ۴ نشان داده می شود، null را از منو بالا رو Component انتخاب کنید. مقدار برگشتی یک متد اقدام نمایش صفحه بعدی (مؤلفه وب) را ارائه می دهد. وقتی null را برگشت می دهید، صفحه فعلی دوباره ترسیم می شود.



شکل ۴-۱۲: افزودن یک اقدام

ج - Add را کلیک کنید.

۲- هایپرلینک را اضافه کنید.

نشانه گر را زیر خط در محل نمایش شمارش لود قرار دهید.

منو Dynamic > WoHyper link را انتخاب کنید.

تکست پیش فرض یک لینک جدید Hyperlink می باشد. شما می توانید آنرا با انتخاب تکست و تایپ چیزی مناسبتر از آن مثل Refresh Time جایگزین کنید.

۳- متد refresh Time را به عنصر WoHyperlink وصل کنید. متد refresh Time را به صفت WoHyperlink option مقید سازید.

از متد refresh Time در لیست Main به عنصر WoHyperlink درآگ کنید. وقتی دکمه ماوس را رها می کنید، لیست غیر استاندارد صفات آشکار می شود. صفت action را انتخاب کنید تا نشان دهد که می خواهید متد refreshtime هنگام کلیک روی لینک فراخوانی شود.

۴- Wo.Main را ذخیره کنید.

#### ۴-۱۷ افزودن منطق بر متدها

در نهایت، متد Refresh Time را با استفاده از X Code اصلاح نمائید بنابراین همانطور که در لیست ۶ نشان داده می شود، متغیر Load Count در هر بار استنتاج شدن افزایش می یابد.

- لیست ۶- پیاده سازی یک متد action

```
public WOComponent refreshTime() {  
    loadCount++;  
    return null;  
}
```

#### ۴-۱۸ چگونگی عملکرد حفظ وضعیت

اگر برنامه کاربردی قسمت "مثال: نمایش شمارش صفحه" را ساخته و اجرا کردید، و Refresh time را کلیک کردید، کنترلر لود و تایم آپدیت می شود. شمار لود با هر بار نمایش صفحه توسط یک کاربر خاص افزایش می یابد. این با استفاده از شیء session انجام می شود، شیء که ارتباط بین یک برنامه و یک کلانیت خاص را نشان می دهد.

با هر بار کلیک hyperlink توسط کاربر، webobjects شیء Main را ایجاد کرده و آن را از طریق یک شیء Session به مرورگر وب شما مربوط می سازد. هر بار که با کلیک Refresh time با برنامه ارتباط متقابل برقرار می کنید، از همان شیء main استفاده می شود. اگر پنجره مرورگر دیگری را باز کرده و دوباره با استفاده از URL نمایش داده شده در کادر Xcode Run به برنامه وصل شوید، مثال مجزایی از Main خلق شده و به آن پنجره مربوط می شود. از این پس، می توانید به طور انفرادی با هر دو پنجره کار کنید. در واقع نه تنها مثال جدیدی از main خلق می شود بلکه شیء Session جدیدی هم به وجود می آید.

WebObjects مشخص می کند هنگامی ایجاد یک Session جدید مورد نیاز است که URL وارده فاقد Session ID باشد. اولین باری که شما با استفاده از URL نظیر URL لیست ۷ به برنامه وصل می شوید، WebObjects یک Session را ایجاد کرده و آنرا به Session ID و اطلاعات دیگر اختصاص می دهد. این اطلاعات به URL برگشتی به مرورگر شما به همراه صفحه وب اضافه شده تا نمایش داده شود (لیست ۸ را ببینید). وقتی درخواست دیگری را از مرورگر خود - مثلاً با کلیک Refresh time - ارسال می

کنید، WebObjects از Session ID رمز گذاری شده در URL استفاده کرده تا Session پردازش کننده درخواست را موقعیت یابی کند.

برای کسب اطلاعات بیشتر در مورد مدیریت وضعیت با استفاده از برنامه کاربردی و شیء های Session قسمت "استفاده از برنامه کاربردی و شیء های Session" را مطالعه کنید.

لیست ۷- URL که باعث معرفی یک شیء Session می شود.

لیست ۸- URL با آی دی Session

## ۴-۱۹ استفاده از برنامه و شیء های Session

وب ماهیتاً یک وسیله فاقد وضعیت و حالت می باشد. یک سرور وب درخواستی می گیرد، پاسخی ارائه می دهد و بدون آگاهی از درخواست های قبلی همان کاربر، پاسخ را به مرور وب درخواست کننده بر می گرداند.

بهر حال، یک برنامه وب می تواند حالت بین درخواست های همان کاربر را حفظ کند تا تجربه قابل قبول کاربری را ارائه دهد. به عنوان مثال، وب سایت های فراوانی به شما اجازه می دهد با استفاده از کارت خریدی کالاها را خریداری کنید. چنین برنامه هایی محتوای کارت خرید شما را هنگام هدایت وب سایت به خاطر می آورند. WebObjects یک شناسه منحصر به فرد را با هر درخواست وارده رمز گذاری می کند. این شناسه در حفظ وضعیت در سراسر یک وسیله بی حالت دیگری بکار می رود. در حالیکه می توانید اطلاعات را برگردانده و بین مؤلفه های وب جلو بروید، اغلب نیاز بوده وضعیت مشترک بین مؤلفه ها را حفظ کنید. به جای ارسال این اطلاعات از مؤلفه وب به مؤلفه، می توانید آنرا در سطح بالاتر هر برنامه کاربردی یا هر جلسه (Session) ذخیره کنید.

وقتی برنامه ای را با استفاده یکی از الگوهای X Code برنامه وب خلق می کنید، Application Java به پروژه شما اضافه می شود. Application زیر مجموعه ای از Woapplication است. WebObjects یک شیء Applications را در راه اندازی معرفی می کند. همه مؤلفه های وب برنامه شما رابطه ای با این شیء Application دارند ( ) application را به مؤلفه وب فرستاده تا شیء Application را بطور برنامه ای بدست آورید. شیء های Session و برنامه کاربردی هم در مرورگر شیء Builder WebObjects پدیدار می شود از این رو می توانید عناصر پویای خود را مستقیماً در ویژگیهای آنها مقیدسازی کنید.

می توانید متدهای قبلی از Woapplication را override کرده تا رفتار برنامه وب خود را متعارف کنید. مثلاً می توانید متدهای WOApplication را از مقدار اولیه ده استنتاج کرده تا رفتار پیش فرض یک

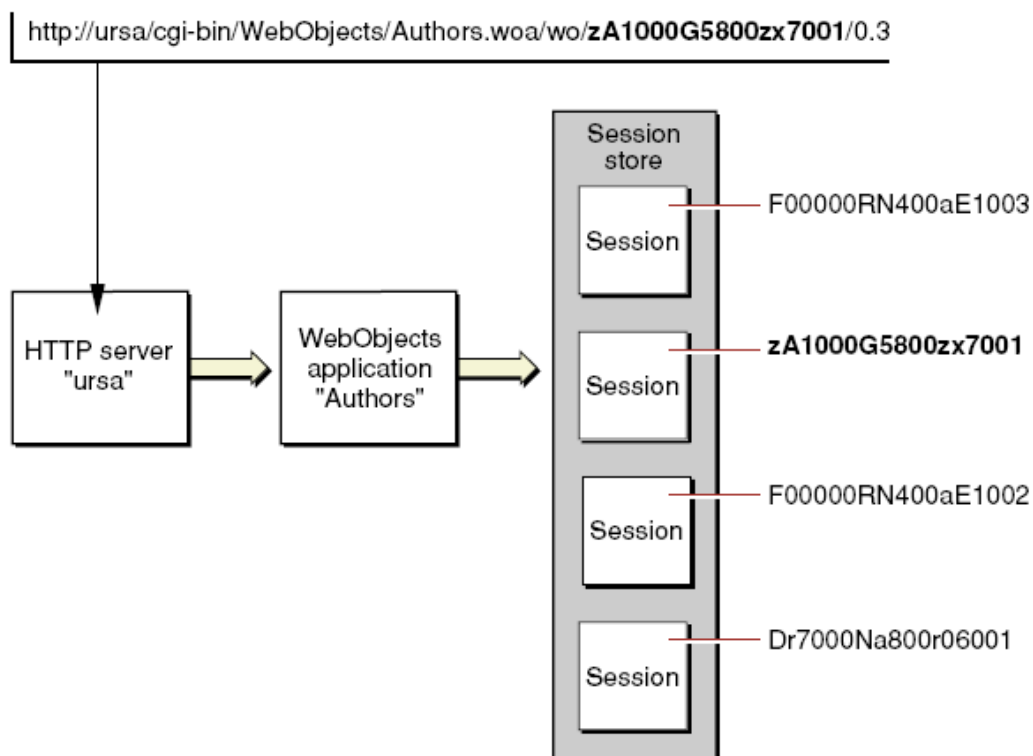
تراک (پیمایش معکوس) را تغییر دهید (برای جزئیات بیشتر "قسمت مدیریت حافظه کش و بک تراک" را مطالعه کنید).

همچنین می توانید متغیرها و متدها را به کلاس Application اضافه کرده تا شیء ها را ذخیره کرده و منطق کاری را اضافه کنید که می خواهید بین مؤلفه های وب و Sessions مشترک باشد. مثلاً، هر شیء صرفاً خواندنی (read-only) و مشترک Sessions را میتوان برای توسعه عملکرد در شیء Application ذخیره کرد.

بهر صورت، اگر می خواهید شیء های سازمانی مشترک باشند، از EOSHaredEditing ontext استفاده کنید نه از کلاس Application برای چگونگی پیکربندی مشخصات واکنشی در مدل خود که شیء های سازمانی را در زمینه ویرایش مشترک قرار می دهد WebObjects enterprise Objects Programming Guide و EO Modeler user Guide را ببینید.

### Session

Session یک دوره زمانی است که یک کاربر در آن زمان با برنامه شما رابطه متقابل دارد. از آنجا که هر برنامه کاربردی می تواند همزمان کاربرهای چندگانه ای داشته باشد. ممکن است session های باز چندگانه ای هم داشته باشد. هر session از دیتای خاص خود و کپی های cach مؤلفه های خاص خود برخوردار بوده که کاربر آنها را درخواست می کند (شکل ۱ را ببینید).



شکل ۴-۱۳: رابطه بین برنامه کاربردی و session



Session عنوان نمونه ای از کلاس session (session.java) در یک پروژه برنامه WebObjects نشان داده می شود. Session زیر مجموعه ای از Wosession است. اساساً، Sesson صرفاً رفتار موروثی داشته اما می توانید متغیرها و متدهای معمولی را اضافه کنید. مثلاً، اگر در حال ساخت یک برنامه خرید آنلاین هستید، Session مکان مناسبی برای ذخیره سازی کارت خرید کاربر خواهد بود زیرا session به یک کاربر خاص پیوند خورده و تا وقتی کاربر از برنامه کاربردی بهره می گیرد ایستادگی می کند. همانطور که در قسمت "حلقه درخواست - پاسخ" شرح داده شده وقتی یک درخواست وارده پردازش می دهد، WebObjects بطور اتوماتیک شیء Session مربوط به کاربری را فعال می کند که درخواست را صادر کرده است.

کلاس Wocomponent متدی برای دست یابی session فعال همزمان را در بر می گیرد. کلاس های Java مؤلفه های وب زیر مجموعه های Wocomponent بوده و WebObjects بطور اتوماتیک Session صحیح را هنگام پردازش درخواست فعال می کند. با ارسال پیام ( ) Session به یک Wocomponent، شیء Session Object را برای کاربر فعلی برگشت می دهد.

مثال: خرید

اگر فروشگاه آنلاینی را پیاده سازی می کنید، کاربران شما به کارت خریدی نیاز داشته تا خریدشان را قبل از بررسی ذخیره کنند. از آنجا که کارت خرید به یک کاربر واحدی تعلق داشته و صرفاً در مدت عمر مفید Session معتبر است، ذخیره سازی کارت خرید در شیء Session منطقی است. مثلاً، نظیر لیست ۱، مثال Per store واقع در `Developer / Examples/ Java webobjects/ Petstore wojava` صفات Cart و Current Account را به کلاس Session اضافه می کند.

لیست یک - کلاس Per store Session

```
public class Session extends WOSession {  
  
    Account account;  
    Cart cart;  
  
    public Session() {  
        super();  
        cart = new Cart();  
    }  
  
    public Account currentAccount() {  
        return account;  
    }  
  
    public void setCurrentAccount(Account newAccount) {  
        account = newAccount;  
    }  
  
    public Cart cart() {  
        return cart;  
    }  
}
```

عمل بک تراک، کش صفحه Clint-side و کش مؤلفه وب سه موضوع دقیقاً مرتبطی است که باعث سردرگمی های فراوانی برای برنامه نویسان برنامه وب می شود. خوشبختانه، webobjects تعدادی مکانیزم ارائه می دهد که در برخورد با مسئله جامع مدیریت وضعیت صفحه به شما کمک می کند. برنامه های پویای وب به دلالتی از جمله ثبات حالت server-side و مدیریت حالت عملی است. HTTP-پروتکل وب- ذاتاً فاقد حالت است. به هر حال، ذخیره سازی وضعیت در سرور برنامه کاربردی مدیریت ثبات در برنامه های وب را عملی می سازد. شیء session در webobjects وضعیتی به خود می گیرد، لیکن به تنهایی مسئول مدیریت وضعیت نیست. شیء session را پیگردی کرده، شیء های WOElement , Wocomponents با شناسه های خاص را flay کرده، مکانیزم های دیگر را برای نگهداری و مدیریت حالت به کار می برد. شیء های Wocomponent وضعیت عناصر پویا و متغیرهای مثال ذاتی خود را مدیریت می کند.

عمل کشی به همراه این متدها نقش مهمی در وضعیت مؤلفه های بصری ایفا می کند. Chaching به کاربر اجازه می دهد صفحه وب قبلاً مشاهده شده (حتی صفحه وب ایجاد شده پویا) را بدون برنامه ای ببیند که نیاز به ایجاد مجدد صفحه دارد. Chaching همچنین نقش تعیین کننده ای در ارائه تجربه خوب کاربر به برنامه وب ایفا می کند. عمل کش به کاربر اجازه می دهد با استفاده از دکمه Back مرورگر وب بک تراک (پیمایش معکوس) کند که اغلب به جای درخواست یک صفحه مشاهده شده قبلی از سوی سرور برنامه کاربردی، لودآنی و سریع صفحات از کش Client-side را تأیید می کند. به هر صورت، از آنجا که پیاده سازی های متنوعی از پروتکل HTTP در سرورهای وب وجود دارد، رفتار بک تراک ثابت نداشته و هنگام توسعه برنامه های کاربردی وب به توجه قابل توجهی نیاز دارد.

علاوه بر کش صفحه Client-side, webobjects همچنین مؤلفه های یک کش Server-side را در حافظه اش ذخیره می کند. اگر به درستی به کار رود، یک مشخصه ارزشمندی است که تجربه کاربر و بازده را توسعه می دهد. اما باید از رابطه بین کش مؤلفه Server-side و کش Client-side و چگونگی تأثیر تناقض و بی ثباتی رفتار بک تراک بر نتیجه در زمان فعال بودن هر دو یا هیچ کدام از مشخصه کش آگاه باشید.

یک مؤلفه وب مجموعه ای از زیرمؤلفه ها و عناصر webobjects است. هنگامی که مرورگر وب صفحه وبی را از برنامه webobjects ذخیره کند، که ایستا HTML یک صفحه ایجاد شده را ذخیره نماید (که مدخل های برنامه های مؤلفه وب نظیر متغیرهای مثال را در برنمی گیرد). در مقابل، عمل کش مؤلفه Server-side وضعیت و تعریف مؤلفه وبی را ذخیره می کند.

کش صفحه Client-side مشخصه ای است که توسط مرورگر وب پیاده سازی شده تا کارایی و تجربه کاربر را توسعه دهد. اگرچه برنامه های webobject اساساً صفحات وب پویا را منتشر می کند. اما وب سایت های فراوانی برای صفحات ایستا به کار می رود. آنها به سرعت سایت های پویای مبتنی بر محتوا تغییر نمی کنند.

به عنوان نمونه، وب سایتی را در نظر بگیرید که داستان های خبری و مقالات دیگر را منتشر می کند. اگر چه سر صفحه سایت احتمالاً دفعات اندکی در هر روز تغییر می کند، احتمالاً در دقیقه های اندکی که کاربر متوسط صرف مرور کردن عناوین و خواندن مقالات کمی می نماید، تغییر نمی یابد.

با فعال بودن کش صفحه Client-side، سر صفحه وب سایت خبری در کامپیوتر کلانیت و بر اساس اولین بازدید در حافظه کش ذخیره می شود. صفحه اول می تواند بزرگ بوده و تصاویر، بنرهای تبلیغاتی و متن داشته باشد. کاربر می تواند مقاله ای را انتخاب کرده، بخشی از آن را بخواند و از طریق URL های

موجود در اولین مقاله به مقالات دیگری دسترسی داشته باشد. سپس، با بازدید پنج یا شش صفحه درون وب سایت، کاربر می تواند به صفحه اصلی بک تراک کند. از آنجا که محتوای آن صفحه احتمالاً در زمانی تغییر نمی کند که کاربر صرف بررسی پنج یا شش صفحه می کند، صفحه را باید دوباره از کش محلی لود کرد.

بنابراین، مرورگر وب، به جای درخواست و دانلود مجدد صفحه اصلی از سرور وب- آن را از کش محلی بازیابی می کند در حالی که از یک ramd trip در طول نت ورک تا سرور وب اجتناب می کند. در این مورد، عمل کش صفحه از یک تابع دوستانه و حساس استفاده می کند.

حالا، مورد یک فروشگاه آن لاین را در نظر بگیرید: کاربری کالاهایی را برای خرید انتخاب کرده و آنها را به سبد خرید اضافه می کند. مشاهده یک صفحه وب کش نشان گر سبد خرید به طور کلی ایده مناسبی برای کاربران محسوب نمی شود همچنان که احتمالاً به روزترین اطلاعات را در بر ندارد.

اگر عمل کش صفحه Client-side فعال باشد، به هر صورت این یک احتمال حقیقی به شما می آید. Webobjects تعداد مکانیزم های مربوط به مسائل بک تراک و کش Client-side را فراهم می آورد. اولین چیزی که باید استفاده کنید یک flay در شیء Application بوده که با استفاده از متد set page اولین چیزی که باید استفاده کنید یک flay در شیء Application بوده که با استفاده از متد Refresh on back track Enabled (com. Webobjects. Appserver) wo Application را تنظیم می کند. وقتی page Refresh True on back track Enabled است، تعدادی هدرهای HTTP به هر پاسخ ارائه شده توسط برنامه webobjects اضافه شده تا کش صفحه Client-side را غیر فعال سازد. جدول ۴-۲ این هدرها و مقادیر آنها را نشان می دهد.

جدول ۴-۲ هدرهای پاسخ HTTP که کش صفحه Client-side را غیر فعال می کند.

Header	Value
date	The time the response page was generated.
expires	The time the response is to expire. (Same as date.)
pragma	no-cache
cache-control	private, no-cache, no-store, must-revalidate, max-age = 0

برای کسب اطلاعات بیشتر در مورد هر یکی از این هدرها، قسمت ۹-۱۴ مشخصات HTTP 1.1 را ببینید. ویژگی page Refresh on back track Enabled بر همه پاسخ های ارائه شده توسط یک برنامه کاربردی تأثیر می گذارد. اگر می خواهید رفتار را به پاسخ خاصی محدود کنید، متد disable Client

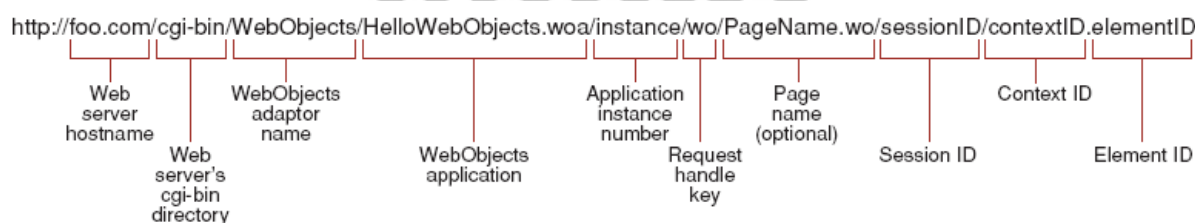


Caching شیء WResponse (com. Webobjects. Appserver) را استنتاج کنید. WReponse هم متدهای setHeader, setHeaders را دربرمی گیرد که به شما اجازه می دهد هدرهای HTTP را به وضوح برای پاسخ خاصی تنظیم کنید.

هنگامی که مرورگر وب صفحه پاسخی با هدرهای نشان داده شده در جدول ۱ دریافت می کند، نباید صفحه را به کش محلی خود اضافه کند و باید صفحه را به محض نمایش نامعتبر کند. به عبارت دیگر، وقتی کاربران برای بازیابی صفحات مشاهده شده قبلی بک تراک می کنند، مرورگر وب باید صفحه پاسخ را از سرور برنامه درخواست کند. به هر حال، همچنان که در قسمت "رفتار بک تراک مرورگر وب" نشان داده می شود، همه مرورگرهای وب این پروتکل را دنبال نمی کنند. اولین دفعات اندکی که کاربر به صفحات مشاهده شده قبلی بک تراک می کند. اکثریت مرورگرهای وب هدرهای HTTP را نادیده گرفته و صفحه ذخیره در کش را رندر می کنند.

وقتی مرورگر وب نیازمند Refresh کردن یک صفحه منقضی می باشد، درخواست را به سرور برنامه کاربردی ارسال می کند که کش server-side به آن دسترسی داشته تا مجدداً صفحه را بازسازی کند (برای کسب اطلاعات بیشتر پیرامون کش server-side قسمت "کش صفحه server-side" را ببینید). "پیام های حله درخواست- پاسخ" سنکرون سازی، عمل و پاسخ. هنگام پردازش یک صفحه refresh، یک برنامه کاربردی فازهای سنکرون و عمل را طی نمی کند؛ آن صرفاً فاز پاسخ را سپری می کند. بنابراین، چگونه یک برنامه کاربردی می داند صرفاً فاز پاسخ (فقط برگشت از صفحه پاسخ ذخیره شده در کش سرور، به جای ایجاد دوباره آن) را انجام دهد؟

webobjects هر یک از پاسخ های یک زمینه ID را ارزیابی می کند. با هر بار درخواست صفحه خاص از سرور برنامه در طی session توسط یک مرورگر وب Context ID تا ۱ افزایش می یابد. مثال خاصی از WComponent متناظر را مشخص می کند. (شکل ۱ عناصر URL webobjects را نشان می دهد). مخصوصاً یک برنامه کاربردی دورترین مؤلفه WComponent یک Context ID را تعیین کرده که هر بار آن مؤلفه، بخشی از یک پاسخ می باشد. بنابراین اگر همان مؤلفه، زمان های چندگانه ایجاد شده پویا باشد، هر نمونه صفه (هر پاسخ) یک ID منحصر به فرد زمینه ای تعیین می شود.



شکل ۴-۱۴: ساختار یک URL عمل مؤلفه

#### ۴-۲۱ کش تعریف مؤلفه server-side

وقتی مؤلفه وب برای اولین بار در دسترس قرار می گیرد، تعریف آن در حافظه کش server-side قرار می گیرد. درخواست های متوالی همان مؤلفه از تعریف ذخیره شده در کش استفاده می کند. استفاده از کش مؤلفه کارایی را ارتقاء بخشد زیرا برنامه صرفاً یک بار در طی عمر مفید برنامه تعریف مؤلفه را جستجو می کند. شما می توانید کش مؤلفه وب را در سطح برنامه و سطح مؤلفه کنترل کنید. می توانید یک خط مشی کش را برای برنامه (یا فعال یا غیرفعال) همه مؤلفه ها تنظیم کنید، همچنین می توانید چنین خط مشی ای را در مؤلفه های خاصی باطل کنید.

برای تنظیم خط مشی Caching یک برنامه کاربردی یا مؤلفه وب، به ترتیب از متد wo Application set ching Enabled یا WComponent استفاده کنید. ارسال true به عنوان آرگومان عمل کش تعریف مؤلفه وب را فعال می کند در حالی که ارسال Flash آن را غیرفعال می نماید.

#### ۴-۲۲ کش صفحه server-side

علاوه بر کش تعریف مؤلفه، برنامه های webobjects همچنین پاسخ های ارسالی به کلانیت را در حافظه کش ذخیره می کند. وقتی یک صفحه اخیراً ایجاد شده از سرور برنامه درخواست می شود، Context webobjects ID صفحه درخواستی با Context ID صفحات موجود در کش خود را بررسی می کند. اگر تطبیقی و هماهنگی را بیابد، فاز پاسخ حلقه درخواست پاسخ را اجار می کند. این امر پاسخی را برگشت می دهد که محتوای آپدیت شده و جدید Context ID از استنتاج فاز پاسخ حلقه درخواست - پاسخ را دربردارد (ارتباطات پویا مجدداً در فاز پاسخ حل می شوند).

به طور پیش فرض، سرور برنامه webobjects کش صفحه را برای هر session حفظ و نگهداری می کند. هر صفحه ای که کاربر به آن دسترسی دارد به کش صفحه session اضافه می شود. وقتی کاربری بک تراک می کند، به URL دسترسی می یابد یا معیاری از صفحه ای را انتخاب می کند که کش شده اما در کش محلی به سر می آید، مرورگر وب یک نسخه بازسازی شده آن صفحه را از سرور برنامه درخواست می کند. کش صفحه server-side منابع را در حالی حفظ می کند که نتیجه صفحات ایجاد شده قبلی را خراب می کند. سپس صفحه ای که کاربر به آن بک تراک می کند دیگر حافظه کش نمی باشد و webobjects یک صفحه error را برگشت می دهد.

اگر کش صفحه server-sied (با ارسال 0 به متد wo Application setpaye Cachsize) را غیر فعال کنید، برنامه فرض می کند که می خواهید ثبات وضعیت مؤلفه معمولی را ارائه دهید به جای آنکه به پشتیبانی ذاتی webobjects استناد کنید.

غیرفعال سازی کش مؤلفه به این معناست که شیء های جدید WOComonents با هر چرخه حلقه درخواست- پاسخ معرفی می شوند (یعنی، هر درخواست برای مؤلفه ای نمونه جدیدی از آن مؤلفه را به وجود می آورد) حتی برای درخواست های عمل مؤلفه که صفحه استنتاج را برگشت می دهند. یعنی هیچ مقدار متغیر غیرپیش فرض نمونه با هر چرخه بعدی حلقه درخواست- پاسخ دور انداخته نمی شود. در برنامه های گسترده، این افزایش و تکرار می تواند کارایی را به تأخیر بیندازد.

webobjects همچنین کش صفحه همیشگی فراهم می کند که برای ذخیره سازی زیر مؤلفه هایی چون میله های چند رسانه ای یا هدرهای صفحه یا هنگام به کارگیری مجموعه های قالب سودمند می باشد. شما باید به وضوح مؤلفه هایی را با استفاده از متد save page In permanent Cach (com. Wosession Webobjects. Appserver) اضافه کنید.

رفتار بک تراک مرورگر وب برای درک بهتر مفاهیم بک تراک، عمل کش Client- side و عمل کش تعریف مؤلفه کارهای مشروح در قسمت های ذیل را انجام دهید. برنامه وبی که در قسمت "ایجاد مؤلفه های وب" توسعه دادید یا هر برنامه ساده وب دیگری را باز کنید. در Main.java متدی به نام Outgoing Headers وارد کنید:

```
public String outgoingHeaders() {  
    return context().response().headers().toString();  
}
```

آن هدرهایی را می گیرد که به هر شیء در حال پیشرفت WOResponse ضمیمه شده است. برای مشاهده این هدرها، متد Sleep را در کلاس Main باطل می کند. از این رو، آن هدرها را در کنسول چاپ می کند:

```
public void sleep() {  
    System.out.println("<Main.sleep> headers=" + outgoingHeaders());  
}
```

برنامه کاربردی را بسازید و اجرا کنید. شما باید خروجی ای مشابه این در کنسول ببینید:

```
Welcome to WebApp!  
[2003-01-08 17:53:56 PST] <main> Opening application's URL in browser:  
http://17.203.33.19:8888/cgi-bin/WebObjects/WebApp.woa  
[2003-01-08 17:53:56 PST] <main> Waiting for requests...
```

```
<Main.sleep> headers={cache-control = ("private", "no-cache", "no-store",  
"must-revalidate", "max-age=0");  
expires = ("Thu, 09-Jan-2003 01:53:54 GMT"); date = ("Thu, 09-Jan-2003 01:53:54  
GMT"); pragma = ("no-  
cache"); content-type = ("text/html"); }
```

هدر expires با زمان ایجاد مؤلفه تنظیم می شود، بنابراین وقتی مرورگر وب webjage را دریافت می کنید، به تازگی در کش مرورگر وب به پایان رسیده است. این هدرها (به غیر از content-type) وقتی به پاسخ پیوست می شود که متد true wo Application is page Refresh on back track Enabled را برگشت می دهد. این امر به طور پیش فرض انجام می شود.  
در Application. Java ویژگی page Refresh on back track Enabled را به flash در سازنده تنظیم کنید:

```
public Application() {  
    super();  
    System.out.println("Welcome to " + this.name() + "!");  
    setPageRefreshOnBacktrackEnabled(false);  
}
```

برنامه را بسازید و اجرا کنید. باید نتیجه ای مشابه نتیجه ذیل را در کنسول مشاهده کنید:  
توجه کنید که هدرهای غیرفعال کننده عمل کش Client-side در پاسخ ایجاد نمی شوند.

```
Welcome to WebApp!  
[2003-01-08 17:57:15 PST] <main> Opening application's URL in browser:  
http://17.203.33.19:8888/cgi-bin/WebObjects/WebApp.woa  
[2003-01-08 17:57:15 PST] <main> Waiting for requests...  
<Main.sleep> headers={content-type = ("text/html"); }
```



بک تراک استاندارد صفحات وب

پس، چگونه ویژگی wo Application page Refresh on back track Enabled بر عمل بک تراک کاربر تأثیر می گذارد؟ نیاز بوده مقداری که بیشتر اضافه کنید تا پیگیری کند webobjects در پشت صحنه چه انجام می دهد. سازنده در کلاس Main را اصلاح کنید تا اینگونه شوند:

```
public Main(WOContext context) {  
    super(context);  
    System.out.println("<Main> context ID="+ context().contextID());  
}
```

هر باری که نمونه ای از Main ایجاد می شود، این که Context ID شیء WOREponse مربوط به نمونه جدید را در خروجی ارائه می دهد. این به شما اجازه می دهد ببینید چه موقع اقدامات کاربر مانند کلیک هایپرلینک Refresh در صفحه وب یا کلیک دکمه Back مرورگر وب نمونه جدیدی از مؤلفه Main ایجاد می کند. در حالی که این اطلاعات مفیدی به شمار می آید، شاید شما بخواهید بدانید که چه موقع اقدام یک کاربر باعث می شود برنامه کاربردی صفحه جدید پاسخی را به مرورگر وب کلانیت ارسال کند. می توانید این را با افزودن کد مشابه به متد refresh time پیگیری کنید:

```
public WOComponent refreshTime() {  
    System.out.println("<Main.refresh> context ID=" + context().contextID());  
    loadCount++;  
    return null;  
}
```

حالا، متدهای sleep، Outgoing Headers را حذف نموده و برنامه را بسازید و اجرا کنید. refresh time را سه بار کلیک کنید. این امر Context ID افزایشی نمونه Main را از طریق نمونه ای چاپ می کند که شما آن را هدایت می کنید. با کلیک Refresh time برنامه متد refresh time را استنتاج می نماید که Context ID پاسخ صادر شده را در کنسول به شکل خروجی ارائه می دهد:

```
Welcome to WebApp!  
[2003-01-08 18:56:18 PST] <main> Opening application's URL in browser:  
http://17.203.33.19:8888/cgi-bin/WebObjects/WebApp.woa  
[2003-01-08 18:56:18 PST] <main> Waiting for requests...  
<Main> context ID=0  
<Main.refreshTime> context ID: 1  
<Main.refreshTime> context ID: 2  
<Main.refreshTime> context ID: 3
```

حالا، دکمه Back مرورگر خود را سه بار کلیک کنید. توجه کنید که هیچ چیزی در کنسول چاپ نمی شود زیرا وقتی page Refresh on back track Enabled روی flash تنظیم می شود. عمل بک تراک باعث ایجاد درخواستی در برنامه نمی شود؛ صفحه با استفاده از کپی در کش مرورگر به سادگی رندر می شود. همین طور انتخاب معیار یک صفحه کش شده در مرورگر وب به درخواستی در برنامه منتهی می شود.

وقتی page Refresh on back track Enabled روی true تنظیم می شود، اگرچه رفتار واقعی در میان مرورگرهای مختلف وب متفاوت است اما عمل بک تراک باید هنگام بک تراک یک کاربر باید درخواستی در برنامه کاربری ایجاد کند.

مرورگرهای وب در Max OSX که از موتور رندر ژکور HTML (نظیر موزیلا و Chimera) استفاده می کند، دقیقاً با مشخصات HTTP مطابقت دارد. کلیک دکمه Back باعث می شود مرورگر نسخه آپدیتی یک صفحه وب expried را درخواست کند. مرورگرهای وب نظیر Omniweb, Internert Explorer به طور متفاوت رفتار می کند: کلیک های محدود اول (دو تا سه کلیک، براساس مرورگر) دکمه Back دوباره صفحه را از کش لود می کند. کلیک های متوالی باعث می شود مرورگر درخواستی را به برنامه ارسال کند.

دقت کنید وقتی مرورگر نسخه آپدیتی webpage را از برنامه درخواست می نماید، کنتور لود پیج کاهش نمی یابد اما تایم آپدیت می شود.

شما باید برنامه خود را در پیکربندی های زیادی تست کنید تا مطمئن شوید که تجربه مناسب کاربر را ارائه می دهد.

#### ۴-۲۳ رد کش server- side

برنامه webobjects صرفاً پاسخ صفحه ایجاد شده قبلی را وقتی به طور دستی اعمال می کند که کش صفحه server- side فعال است که به طور پیش فرض فعال می باشد. زمانی که این ویژگی غیرفعال باشد، عبارت print In در سازنده کلاس Main (برنامه وب که قبلاً شرح داده شد) با هر بار کلیک لینک Refresh Time استنتاج می شود. این نشان می دهد که برنامه کاربردی به جای برگشت دادن شیء فعلی Main را با هر بار استنتاج متد Main refresh time معرفی می کند.

سازنده کلاس Main را با اضافه کردن فراخوانی set page Cache size اصلاح کنید:

```
public Application() {  
    super();  
    System.out.println("Welcome to " + this.name() + "!!");  
}
```

```
setPageRefreshOnBacktrackEnabled(true);  
setPageCacheSize(0);  
}
```

برنامه را بسازید و اجرا کنید، بعد از سه بار کلیک Refresh Time شما باید خروجی کنسول ذیل را ببینید:

```
Welcome to WebApp!  
[2003-01-08 20:31:58 PST] <main> Opening application's URL in browser:  
http://17.203.33.19:8888/cgi-bin/WebObjects/WebApp.woa  
[2003-01-08 20:31:57 PST] <main> Waiting for requests...  
<Main> context ID=0  
<Main> context ID=1  
<Main.refreshTime> context ID: 1  
<Main> context ID=2  
<Main.refreshTime> context ID: 2  
<Main> context ID=3  
<Main.refreshTime> context ID: 3
```

توجه نمایید قبل از آنکه مدت Refresh Time اجرا شود، سازنده موجود در کلاس Main با هر بار کلیک Refresh Time استنتاج می شود. نمونه Main در طی هر چرخه حلقه درخواست- پاسخ به وجود می آید. همچنین دقت کنید که کنتور نمایش صفحه افزایش نمی یابد. نتیجه اولیه غیرفعال سازی عمل کش صفحه server-side این است که مقادیر متغیرها در مؤلفه ها بعد از ایجاد هر پاسخی از بین می رود. به جای رد کش server- side می توانید از مدت set page Cache size wo Application در تعریف تعدادی از نمونه های مؤلفه استفاده کنید که برنامه کاربردی آن مؤلفه ها را در کش خود نگه می دارد. به عنوان مثال، اگر می خواهید وضعیت بین چرخه های حلقه درخواست- پاسخ را حفظ کنید (یعنی مطمئن شوید که وضعیت بین اقدامات کاربر منتقل می شود)، page Cache size را در ۱۰ تنظیم کنید.

```
public Application() {  
    super();  
    System.out.println("Welcome to " + this.name() + "!");  
    setPageRefreshOnBacktrackEnabled(true);  
    setPageCacheSize(10);  
}
```

شکل ۴-۱۵ نشان می دهد وقتی کاربری خیلی دور بک تراک کند (دیگر صفحه در کش وجود ندارد)، برنامه کاربردی صفحه را به مرورگر وب ارسال می کند.



شکل ۴-۱۵ بک تراک صفحه error

شما می توانید error page دریافتی کاربران را با پیاده سازی متد `handlePageRestorationErrorInContext` در کلاس `Appication` متعارف سازی کنید:

```
public WResponse handlePageRestorationErrorInContext(WOContext aContext) {  
    WOComponent nextPage;  
    nextPage = (Error)pageWithName("Error", aContext);  
    return nextPage.generateResponse();  
}
```

صفحه ای در این لیست کد از یک مؤلفه وب به نام `Error` معرفی می شود که شما باید آن را بسازید. محتوای مؤلفه کاملاً مناسب شما بوده، اما باید نام برنامه، نام کمپانی شما، و پیام دوستانه ای را در برداشته باشد که به کاربر بگوید چیزی اشتباه است و راه هایی را پیشنهاد می کند که آنها می توانند به عملیات نرمال برگردند.



### تاریخچه اصلاح سند

این جدول تغییرات راهنمای برنامه نویسی برنامه های webobject وب را تشریح می کند.

تاریخ	نکته ها
۲۰۰۶/۱/۱۰	بازنویسی کامل برای شرح چگونگی ایجاد برنامه های webobject مبتنی بر HTML. این پرونده قبلاً "برنامه های وب" نامیده شده بود.
۲۰۰۳/۴/۱	تصحیح تعریف متد adduser در "نمایش شرطی با عناصر Woconditional" تا وقتی personname یا favoritefood null می باشد، Null pointer Exception ایجاد شده را جابجا می کند. همین تصحیح در variables/ user Entry/ main. Java projects/ Instantce- using- 1- Input- user- Managing- اعمال شد.
	تغییر مرجع نادرست "WORepetition to wofrom" ایجاد مؤلفه "Comfrim Author Delete"
۲۰۰۳/۲/۱	اضافه کردن فصل "مدیریت کش و بک تراک" اعمال تغییرات ویرایشی شامل تغییر web به web و webpage به webpage, keypath به keypath. ۱۴
۲۰۰۲/۹/۱۱	تغییر نام مدارک به webobjects داخلی: برنامه های وب
	نمونه های پروژه حالا در ۱۵
	تجدید نظر 5.2 webobjects
۲۰۰۱/۵/۱۱	انتشار مدارک به عنوان webobjects داخلی: کشف webobjects برای HTML.

## حرف آخر:

### نتیجه گیری و ارائه پیشنهادات

آنچه بیان شد ضرورت ارزیابی اطلاعات بازیابی شده در وب و گوشه ای از معیارهای آن بود که تنها به عنوان الگویی بیان گردید. این معیارها با توجه به نوع کاربران و هدف از جستجو اولویت می یابند و لازم است به آنها تنها به دید الگویی برای ارزیابی منتقدانه و هوشمندانه اطلاعات توجه شود. کتابخانه ها و مراکز اطلاع رسانی با ایجاد و تقویت مهارت های اطلاع یابی و تفکر انتقادی در کاربران به آنها می آموزند که از میان انبوه اطلاعات بازیابی شده گهر را از زباله تشخیص دهند و با استناد به منابع معتبر بر اعتبار فعالیتهای تحقیقاتی خویش بیفزایند. آموزش این مهارتها در گرو برنامه ریزی اصولی و مناسب این مراکز است. کتابخانه ها و مراکز اطلاع رسانی از طریق آموزشهای مجازی و هدفمند قادر به یاری رسانی به کاربران می باشند و نباید در عصر دیجیتال نقش تعلیم دهندگی خویش را از یاد ببرند. مهارت های اطلاع یابی و ایجاد تفکر انتقادی در کاربران کتابخانه از طرق مختلف انجام می پذیرد از روشهای معمول آن برگزاری کلاسهای آموزشی حضوری، تهیه و توزیع بوشورهای آموزشی و ارائه آموزشهای مجازی در محیط وب است. ارائه این آموزش ها مستلزم آن است که کتابداران و اطلاع رسانان ابتدا خود به مهارت ارزیابی و قدرت تفکر انتقادی مجهز باشند. امید آن است که با همت کتابداران و اطلاع رسانان در عصر دیجیتال کاربرانی تربیت شوند و تفکر انتقادی را به صورتی اصولی در خویش تقویت کرده باشند و در مفهوم کاربرانی مستقل به بازیابی اطلاعات بپردازند. جامعه اطلاعاتی زمانی تحقق می یابد که تک تک شهروندان به عنوان پاره های جامعه از دیدی تحلیلی و منتقدانه برخوردار باشند و مبنای تصمیم گیری هایشان اطلاعات صحیح و روزآمد باشد.

## منابع

- حری، عباس (۱۳۸۲). اطلاع رسانی: نگرش ها و پژوهش ها. تهران: نشر کتابدار.
- حسن زاده، محمد. "تعامل حرفه کتابداری و اطلاع رسانی با فرایند جهانی شدن". [پیوسته]:  
<<http://www.hassanzadeh.netfirms.com/myart/globinfart.doc>>
- دیانی، محمد حسین و عبدلحمید معروف زاده (۱۳۷۰). مبانی مرجع: فنون و روشهای پرسش کاوی و پاسخ یابی از منابع کتابخانه. مشهد: آستان قدس رضوی.
- کوک، آلیسون (۱۳۸۲). راهنمای یافتن اطلاعات با کیفیت در اینترنت. ترجمه مهدی خادمیان. مشهد: چاپار.
- لی، یان مال (۱۳۷۶). اصول آموزش مهارتهای اطلاع یابی. ترجمه فاطمه سبکروح و آذر عمرانی. تهران: واژه پرداز.
- Barker, Jon (2004) "Finding Information on the Internet: A Tutorial University of California".  
Available at:  
<<http://www.lib.berkeley.edu/TeachingLib/Guides/Internet/Evaluate.html>>
- Case, Ronald. (2003) "Making Critical Thinking An Integral part of Electronic Research". School Library in Canada. 22(4).  
Available at: <<http://search.epnet.com/login.aspx>>
- Engle, Micheal. (2004) "The Seven Steps of the Research Process". Cornell University Library.  
Available at: <<http://www.library.cornell.edu/olinuris/ref/research/skill1.htm%20-%20bibliog>>
- Smith, Alistair. (1997) "Criteria for evaluation of Internet Information Resource "The Public-Access Computer Systems Review. 8(3).  
Available at: <[http://www.vuw.ac.nz/staff/alastair\\_smith/evaln/index.htm](http://www.vuw.ac.nz/staff/alastair_smith/evaln/index.htm)>
- Smith, Marshall L. (2001) "Critical Thinking and the Web". Darlene Lynch and Robert F. Vernon, Journal of Social Work Education, 37(2): p381-387.  
Available at: <<http://search.epnet.com/login.aspx>>
- Standler, Ronald B. (2004) "Evaluating Credibility of Information on the Internet".  
Available at: <<http://www.rbs0.com/credible.pdf>>
- Tilloston, Joy. (2003) "A Portrait of the Audience for Instruction in Web Searching: Result of a Survey Conducted at Tow Canadian Universities". CJLIS/RCSIB. 27(1): p3-24.