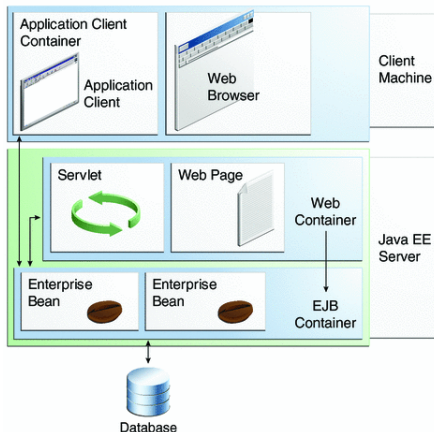


- Create a simple todo list web application
 - Practice using AJAX
 - Clone the starter code from github
 - Complete the functionality of the todo list

Architecture of a Java Web Application



- This diagram¹ shows the essential components of a Java-based web application: client, server, and database

¹E. Jendrock, C.-N. Ricardo, I. Evans, *et al.*, *The Java EE Tutorial*, Release 7. Oracle, 2014.

- GET: requests data from a url
- POST: submit data to be processed at a url
- These methods behave differently in terms of caching, interaction with back button, etc.
- PUT: place some data/document at a url
- GET and PUT must be *idempotent* operations

- *Front-end web development:* Design and coding of client-side HTML, CSS, and Javascript
- Although web browsers are extremely common there is still lots of fragmentation
 - Screen size
 - PC and mobile
 - Operating System
 - Browser
- Dynamic behavior of websites is achieved with Javascript

- jQuery is the most popular Javascript library today
- Free, open source software (MIT License)
- Tools for making scripting easier: selecting elements, adding handlers, ...
- Concise syntax
- Handles differences in browsers



```
// select all first paragraph
document.getElementsByTagName("p")[0].innerHTML = "New text.";

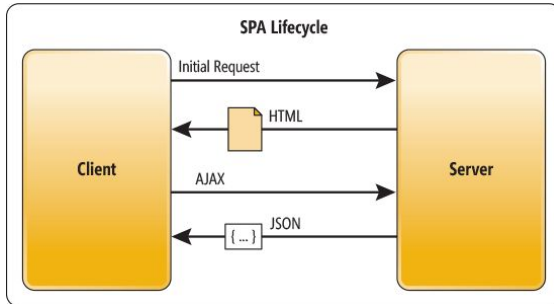
// same query in jQuery
$("p").html("New text.");
```

- These lines of code will each replace the contents of the “first” `<p>` with “New text.”
- The dollar sign is a shorthand for `jQuery()`
- The `jQuery()` function is heavily overloaded, we can pass many different things to it depending on what we want to achieve

- Java EE is a set of specifications and APIs
- There are many different implementations of these specifications, some more complete than others
- Apache Tomcat is a Servlet Container which does not implement the complete JavaEE Spec
 - *TomEE* is a more complete EE server built on top of Tomcat
- For your projects the server has a few important functions
 - Serve up HTML content to the user
 - Serve machine readable content to the browser (enable dynamic pages)
 - Connect to the database when needed

- You might have some static HTML/JS pages that you simply want to make available through your server
- Recall that Java EE Web apps rely on a certain directory structure
- Static HTML Content can be placed in `src/main/webapp/` (we can also put subdirectories under `webapp`)
- JSP pages can be served in the same way
 - Javaserer Pages (JSP) are HTML documents containing special tags `<%...%>` that allow Java code to be embedded
 - The EE server will dynamically convert these pages into servlets

Single Page Application Sequence Diagram



- This sequence diagram shows the basic interaction we would like to achieve
- The user makes an initial request, and subsequent communications with the server happen through *ajax*

- AJAX requests can also be made through the jQuery function
- Syntax is a bit simpler than using xhr directly
- `$.ajax()`
 - Asynchronous Ajax request
 - Pass in the parameters of the request as a Javascript object: url, callback function, type of request, etc.
- Also have convenience methods for Ajax requests: `.load()`, `.get()`, `.post()`

- We have set up a public repository containing a basic Java web application²
- You can clone this project and import to Eclipse as you have done previously
- Make sure you are in the Java EE perspective (upper right corner)

²<https://github.com/marks1024/dynamic-todo-list-exercise-361>

- The sample application allows the user to view a todo list and update the list with a new item
- Your task is to make this a fully functioning todo list
 - Add a new button that will allow the user to clear the entire todo list
 - Add the ability to delete individual items that are currently in the todo list
 - Validate the todo list items using a reasonable regular expression
 - **Bonus** Incorporate templating, as described above, in your solution (see next slide)
- Deploy your application and show to teaching staff when finished
- You will need to edit code on both the front-end and server side

```
// Create a template function
var listTemplate = _.template('<p><%= name %></p>');

// ...
// Use the template function
$('#maindiv').append(listTemplate({name : e}));
```

- A *template* is a special function that performs string interpolation
 - String interpolation: create a string by substituting for placeholder values
- In the example, we use `lodash`³ (a utility library for javascript) to create a template function
- The template function takes a javascript object as an argument

³<https://lodash.com/docs/#template>