Strukturę "Dynamic Web Project 3.0" widzimy po prawej. Składa się ona z:

* **JAX-WS Web Services**

Folder ten zawiera serwisy sieciowe oparte o Java Api for XML. Folder ten nas nie będzie interesował w tym projekcie.

* **Deployment Description: HelloWorld**

Folder ten zawiera wszystkie komponenty, jakie zostały zawarte w naszym projekcie. Wspiera on wyszukiwanie, edytowanie oraz tworzenie nowych komponentów projektu, takich jak np Servlety. Eclipse mapuje katalogi projektu w poszukiwaniu odpowiednich komponentów i segreguje ich do odpowiednich katalogów.

* **Java Resources**

Folder ten zawiera wszystkie komponenty stworzone przez nas, takie jak pakiety, klasy czy interfejsy. To tu będą definiowane wszystkie źródłowe pliki.

* **JavaScript Resources**

Dynamic Web Project w pełni wspiera język JavaScript i tutaj zawarte są komponenty tego języka, które później będzie można wykorzystywać przy tworzeniu stron www.

* **build**

Folder zawiera wszystkie komponenty projektu, które będą tworzone przy buildowaniu projektu. Folder ten nas nie będzie interesował w tym projekcie.

* **webapp**

Folder zawierający pliki konfiguracyjne projektu.

Plik MANIFEST.MF w folderze META-INF zawiera informacje o zależnościach z innych dynamicznych web modułów.

WEB-INF zawiera folder lib, gdzie można dodać wszystkie biblioteki, z których chcielibyśmy w przyszłości korzystać w naszej aplikacji. Folder również zawiera plik web.xml.

To co nas będzie interesować, to głównie foldery **Java Resources -> src**oraz **webapp -> WEB-INF**.

Plik web.xml – tu musimy powiedzieć Tomcatowi gdzie jest w projekcie nasz servlet oraz jak ma go mapować.

W sekcji <servlet> definiujemy nazwę naszego servletu (<servlet-name>) oraz podajemy ścieżkę (<servlet-class>) do klasy reprezentującą nasz servlet (paczka pl.javastart.servlets oraz klasa HelloWorldServlet).

W sekcji <servlet-mapping> ustalamy sposób w jaki tomcat ma mapować poszczególne servlety naszego projektu. Dla servletu "Hello World" link url (<url-pattern>) ustawiłem wartość "/test". Oznacza to, że gdy w przeglądarce wpiszemy "http://[adres serwera]/[nazwa projektu]/test" uruchomimy nasz servlet. <url-pattern> możemy ustawić na cokolwiek.

Serwer jest również zwany **kontenerem webowym**. Przykładem takiego kontenera będzie serwer Apache Tomcat.

**Protokół komunikacyjny HTTP** jest to sposób za pomocą którego w dzisiejszych czasach łączymy się np do stron www przez przeglądarki . Działanie takiego protokołu jest bardzo proste - przeglądarka wysyła żądanie do serwera z prośbą o zasób, a serwer zwraca odpowiedź. Na tej zasadzie będziemy budować nasze aplikacje w Javie EE.