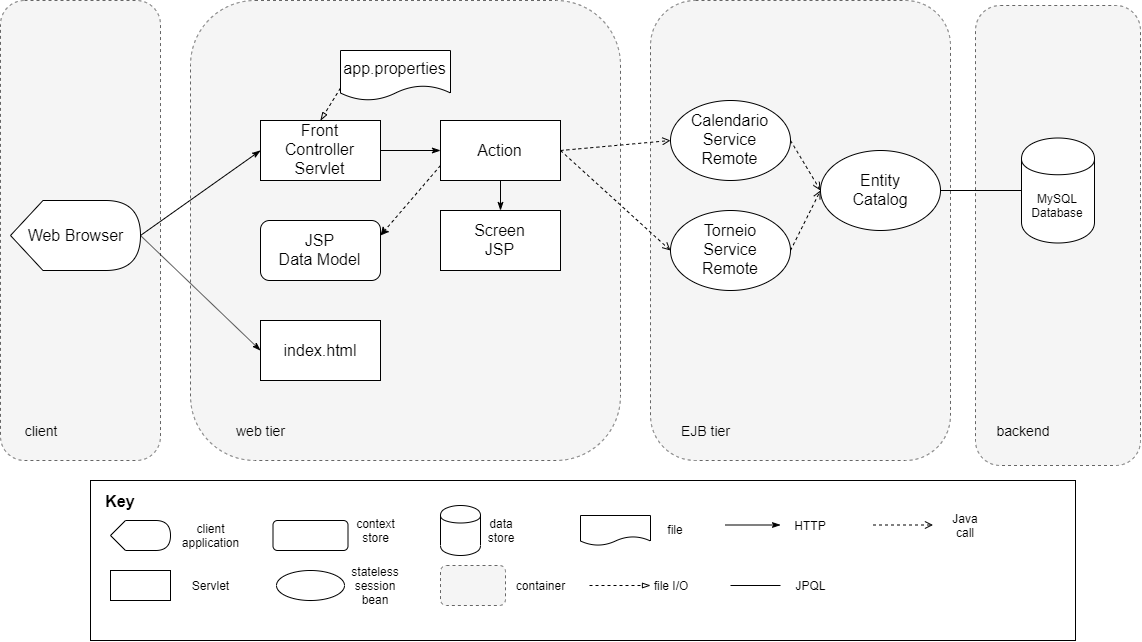
View name: Web client multi-tier C&C view

Primary representation:



Element Catalog:

* Web Browser – this component represents the interface from which the client will view the application. The interface will be mainly exposed by html files and JSP files. The web application itself is run on a WildFly server.
* Front Controller Servlet – this servlet is responsible for mapping all HTTP requests from the web application to their respective action object, from which the input or output data will be handled and displayed. For every action there is a path defined within app.properties file, which is loaded into the servlet upon initialization, thus mapping all action objects to their corresponding paths. Any action that is not mapped by the servlet will be categorized as an unknown action.
* app.properties – this file contains the path mappings to all the available actions for all the use cases implemented in the web client.
* Index.html – this file displays the initial welcome page. It contains 2 URLs which are associated to use cases view player schedule and store match results.
* Action – this component handles the contents of incoming http requests from the web client. The data to and from the database will be sent and retrieved by a specific action depending on the use case in question. It will have access to the 2 main services from within the business logic, which are injected as stateless EJBs. It will also be responsible to forwarding the HTTP request to the corresponding JSP file.
* Screen JSP – these JSP files are part of the web screen. The data required to fill in the fields, if any, is sent via HTTP requests from the action servlets. These requests will contain the data model which contain all the fields necessary for the current JSP, whether it be displaying data, or filling in data to be sent to the backend for further request processing.
* JSP Data Model – this component will be used by a specific screen JSP. It contains all the fields that may be displayed on the loading of the JSP and the fields that will be filled in through the screen JSP, by calling the appropriate action when submitting the data.
* Calendario Service Remote – this component is a remote stateless session bean. It will be accessed by a specific action that takes care of the view player schedule use case.
* Torneio Service Remote – this component is also a remote stateless session bean accessed for the register match result use case, both remotes are injected in the actions and invoked through java calls.
* Entity Catalog – this component will take care of all the database accesses needed through JPQL queries. Whenever a user requests or stores information to the database, the last part of the transaction will always be processed by an entity catalog.
* MySQL Database – this component represents the main datastore from which all the data regarding match scores from the web client will be stored.