Persistence

As recommended by the Chair of the course, the database used in the project will be PostgreSQL. This document is an overview of the different persistence options for each Framework.

General setup (for both Frameworks)

1) Install PostgreSQL - http://www.enterprisedb.com/products-services-training/pgdownload#windows
2) Install pgAdmin http://www.pgadmin.org

1) Persistence in Vaadin

The alternatives for persistence in Vaadin are not very broad. The most popular and well-documented solution is called Vaadin JPAContainer.

Vaadin JPAContainer is an add-on that uses JPA to connect the UI components to the database.

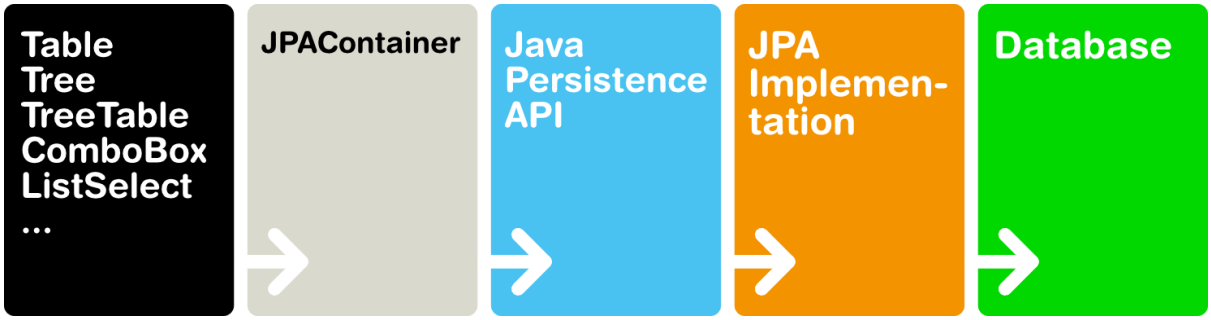

Figure 1 - Persistence in Vaadin (http://vaadin.com/download/jpacontainer-tutorial/)

To get Vaadin JPAContainer running the following JARs are necessary:

- Vaadin JPAContainer
- Java Persistence API 2.0 (javax.persistence package)
- JPA implementation (EclipseLink, Hibernate, …)
- Database driver (H2, MySQL, PostgreSQL…)

The main JPA implementations supported are EclipseLink and Hibernate.

Decisions to make: JPA implementation – EclipseLink versus Hibernate

> **EclipseLink** –It is the reference implementation of JPA 2.0. It's possible to call native SQL functions in your JPQL queries. It has a better lazy loading of entity fields and better exception handling. (Recommended by Vaadin)
>
> **Hibernate** – In general it's more mature, better documented and it has a bigger community.

2) Persistence in GWT

The most popular persistence abstraction alternative for GWT is Hibernate. There are three main forms of integrating GWT with Hibernate:

- **Data Transfer Object** (DTO) - put a light object between the heavy Hibernate object and its data representation that it's taken care about on the client-side.
- **Gilead** - transparent solution to exchange objects between Hibernate and GWT.
- **Dozer** - generate DTOs automatically by reading and parsing XML files.

Comparing the different integration solutions:

DTO It can be designed to be as concise and effective as needed to improve performance.

Gilead is considered invasive for forcing you to extend your domain model entities from the LightEntity class. The project was discontinued, not supporting GWT 2.5 and on.

In **Gilead** and **Dozer** it may become taxing to the user experience when there are larger sets of data to serialize.

3) Conclusion

- **Vaadin -** Recommended persistence configuration: Vaadin JPAContainer (add-on) + EclipseLink (JPA implementation) + PostgreSQL.
- **GWT -** Recommended persistence configuration: DTO (integration strategy) + Hibernate (JPA implementation) + PostgreSQL.