Enabling JPA Support

IntelliJ IDEA provides a dedicated *JPA facet* to support JPA framework. To enable JPA in a module, extend it with the JPA facet. If a new module is being created from scratch, it is possible to enable JPA support in the New Module wizard. If a module already exists, the JPA facet should be added in the Module Settings dialog.

> Only one JPA facet is allowed per module.

To enable JPA support in a new module

1. Create a new Java module from scratch.

   On the first page of the wizard, select **Java Module**, not **Web Module**. (The Web module type is for developing web applications using programming languages other than Java, for example, PHP, or JavaScript, or markup languages.)
2. On the technologies page of the wizard, click the **Java EE Persistence** check box.
3. From the drop-down list, select the default JPA provider that will be used to generate mappings.
4. If you want to import a database schema automatically, select the **Import database schema** check box.
5. Configure a library for the new module. IntelliJ IDEA displays the list of missing archives, and suggests you to either use the libraries, already configured in project, or pick these archives from your file system, and then specify the library name and level.
6. Click **Finish**. The resulting module contains `persistence.xml` file. Further you will need to populate it with persistent units, and map them to data sources.

To enable JPA support in an existing module

1. Open the Module Settings dialog box.
2. With the desired module selected, click + on the toolbar, and select *JPA* from the list of facets available for this module.
3. Click the JPA facet node. The right pane of the dialog box shows the facet options.
4. If you have not configured library for JPA in advance, the facet page displays the list of missing libraries.
   To resolve the problem, click the **Fix** button. In the **Specify Libraries** dialog box, specify whether you would like to use one of the existing libraries, or find the archive in the file system. You can control the target location where the archive will be placed, the library name, and the level on which the library will be created.
5. Create descriptors. To do that, click + (**`Alt+Insert`**) in the **Descriptors** section, and select the desired descriptor type:

   - JPA configuration descriptor `persistence.xml` for defining the classes that should be persisted.
   - JPA mapping descriptor `orm.xml` for specifying persistence using metadata rather than annotations.

   In both cases, specify location where the descriptor files should be stored.
6. Select the default JPA provider from the drop-down list.
7. Click **OK**. The `persistence.xml` file is generated in the `META-INF` directory under the module root. Further you will have to populate it with persistent units, and map them to data sources.

See Also

Concepts:

- Data Sources
- Object-Relational Mapping (EJB, Hibernate and JPA)

Procedures:

- Configuring Module Dependencies and Libraries
- Creating Persistence Units
- Generating Persistence Mappings

Reference:

- Hibernate and JPA Facet Pages
- New Module Wizard

External Links:

- http://java.sun.com/javaee/overview/faq/persistence.jsp

Web Resources:

- Developer Community