

[Download](#)

! The developer Steam4games.it, which makes free games, thanks for the game.Q: How to return a JSON object from an EJB? I am trying to retrieve a JSON object from an EJB. I have the following code: @Stateless public class MySClass { @EJB private MyOtherClass otherClass; public MyObject createObject(){ MyObject obj = new MyObject(); otherClass.doStuff(); ... return obj; } } I would like to see the results of the method otherClass.doStuff() in the browser. What am I missing? I am trying to return an object of type MyObject. What would be the proper way of returning the results? A: Make sure to define a web-app in the EJB, if you don't do that, the FacesServlet that is being used for serving the response will not be able to find and parse the JSP that is returned. If you don't provide anything in the FacesServlet, and no application faces are defined, then a FacesServlet will be created by default. The process will start with the FacesServlet picking up the HTML response, and then it would try to find the JSP defined in the web application and parse it. In your case, it won't be able to find it, because you didn't set up any JSP for your web-app, so it won't be able to find anything. If you don't want to define a web-app, and have the FacesServlet already configured, then make sure that the FacesServlet is started. With an embedded Jetty server, this is done by using the startup element with the parameter server="liferay-portlet". Otherwise, if you have a separate servlet container, you can start it as follows: java -jar mycontainer.jar --server=myserver -startup=com.liferay.portlet.setup.UpgradeAllPortletStubs The startup attribute specifies an additional entry point for the servlet container to look up, and what classes to include. See also: What are the different Jetty server startup modes?

