

---

## Hama USB Gamepad Force Driver for PC joysticks Windows 7 and 8 (S2013)?

[Download](#)

. The Hama Black Steel Pad is a perfect tool for those who want to play games with the guarantee of pure gaming. The great movement of the directional pad and the powerful vibration of the joystick guarantee a perfect quality of gaming. A compact model for players who want the feeling of pressing buttons with their fingers.Q: Send array of objects in response Is there an elegant way to send an array of custom objects back to the client in the response? I'm aware of Jackson and JAX-RS, but they seem to be limited to serializing and deserializing the single object. I'd rather have an elegant solution using REST. I'm comfortable with pure Java and implementing my custom serialization/deserialization. It seems like this should be a simple thing to do, but I've been spending hours searching for a solution, so I figure it has to be documented somewhere. A: Yes, this is a solved problem. It's called JSON and has been a very popular choice for returning data to clients. For example, in JSP/EL you can do: response.setContentType("application/json"); JSONObject json = new JSONObject(); json.put("key1", "value1"); json.put("key2", 2); json.put("key3", "value3"); json.put("key4", new Date()); response.getWriter().write(json.toString()); If you need something more advanced (e.g. nesting) you can use GSON. P.S.: Pay attention when you serialize the request to JSON, otherwise you can end up sending a string representation (i.e. "[[key1]]" instead of {"key1": "value1"}). A: Yes, this can be done with the optional @Produces(MediaType.APPLICATION\_JSON) annotation. Example: @Path("/accounts") public class AccountResource { @GET @Path("/login") @Produces(MediaType.APPLICATION\_JSON) public Response login(@HeaderParam("Username") String username, @HeaderParam("Password") String password) { if (username != null && password != null) { return Response.status(200).entity("ok").build(); }

