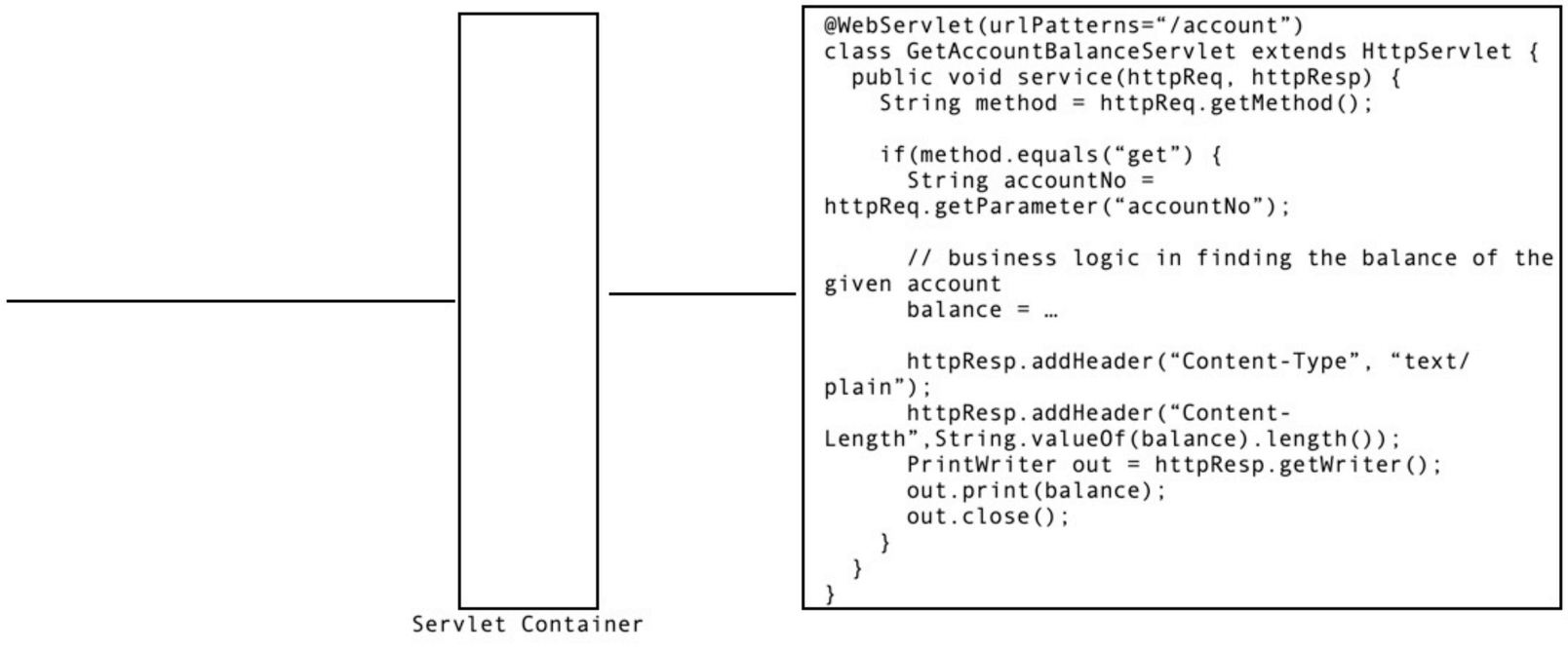
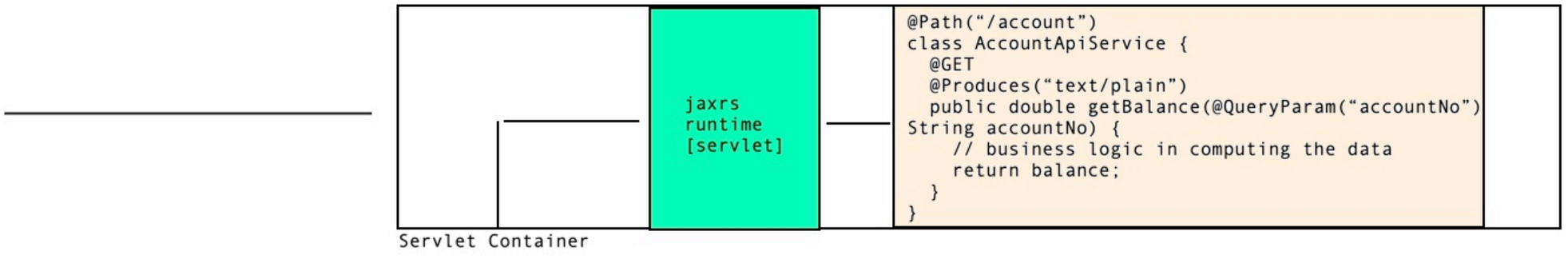




jaxrs api
1. developer implementation in building software application based on the api

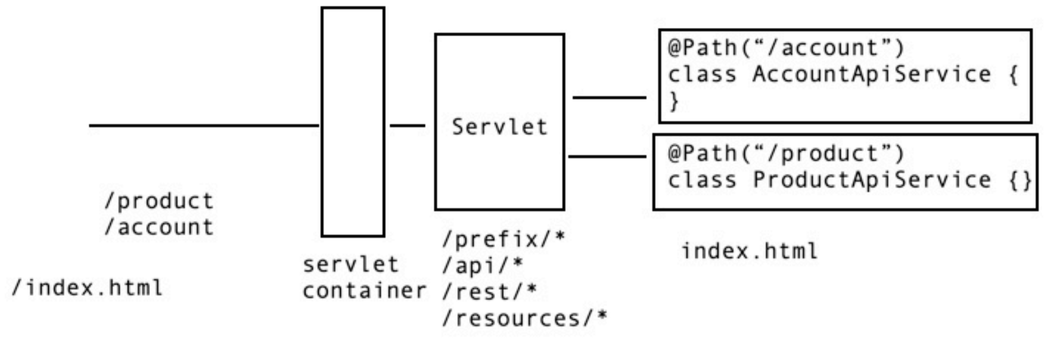
2. runtime implementation
 2.1 receive the httpprotocol request from network
 2.2 invoke the api implementation classes written by developer
 2.3 collect the response from the api class and dispatch to client
server implementation (jaxrs servers)

but implementing jaxrs servers from scratch is waste, because partially we have servlet containers in the place that does half of the logic
1. receive http request
2. dispatch http response
but they cannot invoke jaxrs api class, rather they can invoke servlet class. Now if we can build an extension to the Servlet container, that does the job of taking the request from container and invoke jaxrs api class and collect the output from jaxrs api class and handovers to servlet container then we can eliminate one more jaxrs container

So jaxrs implementation is nothing but an extension to the ServletContainer built using Servlet api.
Every implementation vendor of the jaxrs api should provide one Servlet class that acts as an runtime in invoking the jaxrs api classes as an extension to the ServletContainer.

Servlet api
// interfaces

#2 implementations
1. developer implementation in building application based on that api
2. runtime implementation (servers) (container implementations)
that should receive the request -> dispatch the request to appropriate classes written based on api, and collect the response and send it back to clients



/api/account
/rest/account
/resources/account