

action="/newPatient"

<c:if test="\${errors ne null}">
display errors
</c:if>

patient name: john
dob: 10/13/2001
gender: M: F:
mobile: abc83
email address: john
street address:
city:
state:
zip: hhj
country:
add

new-patient-form.jsp
(source page)

```
@WebServlet("/newPatient")
class PatientSignUpServlet extends HttpServlet {
    public void service(req, resp) {

        PatientForm form = RequestHandler.handle(req,
resp, "patientForm", PatientForm.class,
PatientFormValidator.class, "new-patient");

        // invoke the service to perform business and
persistence operations

        // bind patientId into dataMap
        ViewDispatcher.dispatch(req, resp, "new-
patient-success", map);
    }
}
```

```
class RequestHandler {
    public static Object handle(HttpServletRequest req,
HttpServletResponse resp,String formObjectName
Class<?> formClass, Class<?> validatorClass,
String sourceForm) {
        Validator validator = null;

        Object form = GenericWrapper.wrap(req,
formClass);

        validator = validatorClass.newInstance();
        validator.validate(form);

        if(validator.hasErrors()) {
            Map dataMap = new HashMap();
            dataMap.put(formObjectName, form);
            dataMap.put(formObjectName+".errors",
validator.getErrors());
            ViewDispatcher.dispatch(req, resp,
sourceForm, dataMap);
            return ;
        }
        return form;
    }
}
```

action="/book-appointment"

patient no:
mobile:
doctor:
appointment date/time:
reason:
remarks/comments:

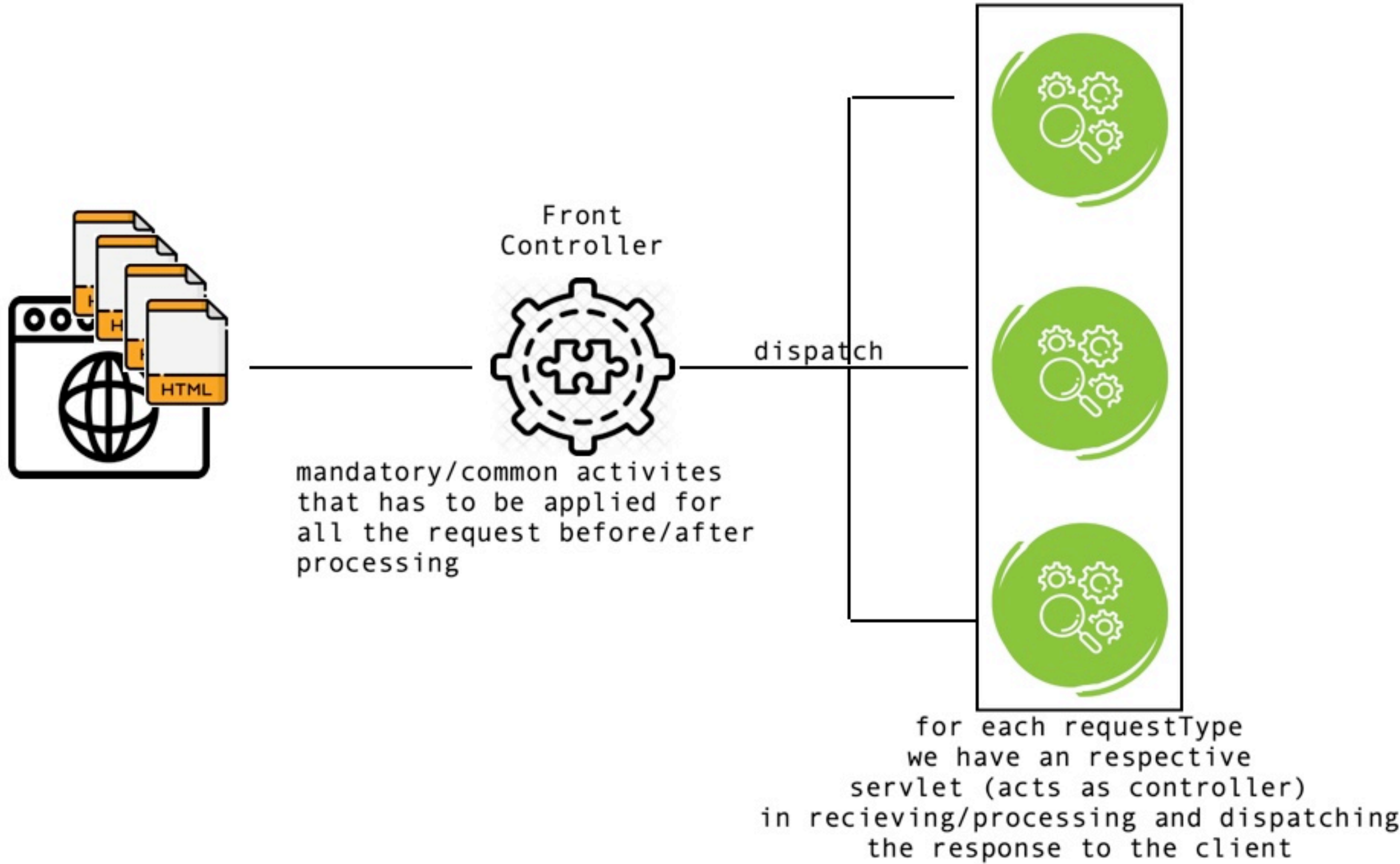
book-appointment.jsp

```
@WebServlet("/book-appointment")
class BookAppointmentServlet extends HttpServlet{

    public void service(req, resp) {
        AppointmentForm form =
RequestHandler.handle(req, resp,
"appointmentForm", AppointmentForm.class,
AppointmentFormValidator.class, "book-
appointment");

        // invoke the service to perform business/
persistence logic
        // collect the data bind to the map
        // forward to the target page

        ViewDispatcher.dispatch(req, resp,
"appointment-info", map);
    }
}
```



```
#mandatory activities (few)
http logging
- source ip
- headers
- req/resp body payload
- status code

ip whitelisting
- allow only the request coming from
#trusted source
#geographical regions
```