

action="/newPatientForm"

patient name:

dob:

gender: ☒ M: ☐ F:

mobile:

email address:

street address:

city:

state:

zip:

country:

new-patient-form.jsp

```
class PatientForm {
    String patientName;
    String dob;
    String gender;
    String mobileNo;
    String emailAddress;
    String streetAddress;
    String city;
    String state;
    String zip;
    String country;
    // accessors
}
```

follow convention in creating the form class: #always the request parameter name should match with attributeName of the form class

```
@WebServlet("/newPatientForm")
class PatientFormRegistrationServlet extends
HttpServlet {
    public void service(req, resp) {
        String patientName =
req.getParameter("patientName");
        String dob = req.getParameter("dob");
        String gender = req.getParameter("gender");
        String mobileNo = req.getParameter("mobileNo");
        String emailAddress =
req.getParameter("emailAddress");
        String streetAddress =
req.getParameter("streetAddress");
        String city = req.getParameter("city");
        String state = req.getParameter("state");
        String zip = req.getParameter("zip");
        String country = req.getParameter("country");

        PatientForm form = new PatientForm();
        form.setPatientName(patientName);
        form.setDob(dob);
        form.setGender(gender);
        form.setMobileNo(mobileNo);
        form.setEmailAddress(emailAddress);
        form.setStreetAddress(streetAddress);
        form.setCity(city);
        form.setState(state);
        form.setZip(zip);
        form.setCountry(country);

        PatientManagementService pmService = new
PatientManagementService();
        int patientNo = pmService.newPatient(form);

        req.setAttribute("patientNo", patientNo);
        req.getRequestDispatcher("/new-patient-
success.jsp").forward(req, resp);
    }
}
```

```
[module]
class PatientManagementService {
    int newPatient(PatientForm patientForm) {
        // business logic
        // invoke dao
    }
}
```

patient no: xxx

patient name:

dob:

gender: ☐ M: ☐ F:

mobile:

email address:

street address:

city:

state:

zip:

country:

update-patient-form.jsp

req.getParameterMap();  
Map  
patientName, john  
dob, 10/01/2001  
gender, M  
mobileNo, 1  
emailAddress, john@gmail.com  
streetAddress,.....  
x, abc

patient no:

mobile:

doctor:

appointment date/time:

reason:

remarks/comments:

book-appointment.jsp

```
class AppointmentForm {
    String patientNo;
    String mobileNo;
    String doctorNo;
    String appointmentDateTime;
    String reason;
    String remarks;
    // accessors
}
```

```
@WebServlet("/bookAppointment")
class BookAppointmentServlet extends HttpServlet {
    public void service(req, resp) {
        String patientNo =
req.getParameter("patientNo");
        String mobileNo = req.getParameter("mobileNo");
        String doctorNo = req.getParameter("doctorNo");
        String dateTime =
req.getParameter("appointmentDateTime");
        String reason = req.getParameter("reason");
        String remarks = req.getParameter("remarks");

        AppointmentForm form = new AppointmentForm();
        form.setPatientNo(patientNo);
        form.setMobileNo(mobileNo);
        form.setDoctorNo(doctorNo);
        form.setAppointmentDateTime(dateTime);
        form.setReason(reason);
        form.setRemarks(remarks);

        AppointmentManagementService amService = new
AppointmentManagementService();
        int appointmentNo =
amService.bookAppointment(form);
        // forward to appointment-details.jsp
    }
}
```

```
[module]
class AppointmentManagementService {
    public int
bookAppointment(AppointmentForm
appointmentForm) {
        // business logic
        // invoke dao
    }
}
```

```
class GenericWrapper {
    Object wrap(HttpServletRequest req, Class<?> clazz) {
        Object obj = clazz.newInstance();

        reqMap.forEach((paramName, paramValue) -> {
            try {
                Field field = clazz.getDeclaredField(paramName);
                String setterName = "set" +
String.valueOf(paramName.charAt(0)).toUpperCase() +
paramName.substring(1);
                Method method = clazz.getDeclaredMethod(setterName,
String.class);
                method.invoke(obj, paramValue[0]);

            } catch (NoSuchFieldException e) {
                // ignore
            } catch (NoSuchMethodException e) {
                throw new RuntimeException(e);
            } catch (InvocationTargetException e) {
                throw new RuntimeException(e);
            } catch (IllegalAccessException e) {
                throw new RuntimeException(e);
            }
        });

        return obj;
    }
}
```