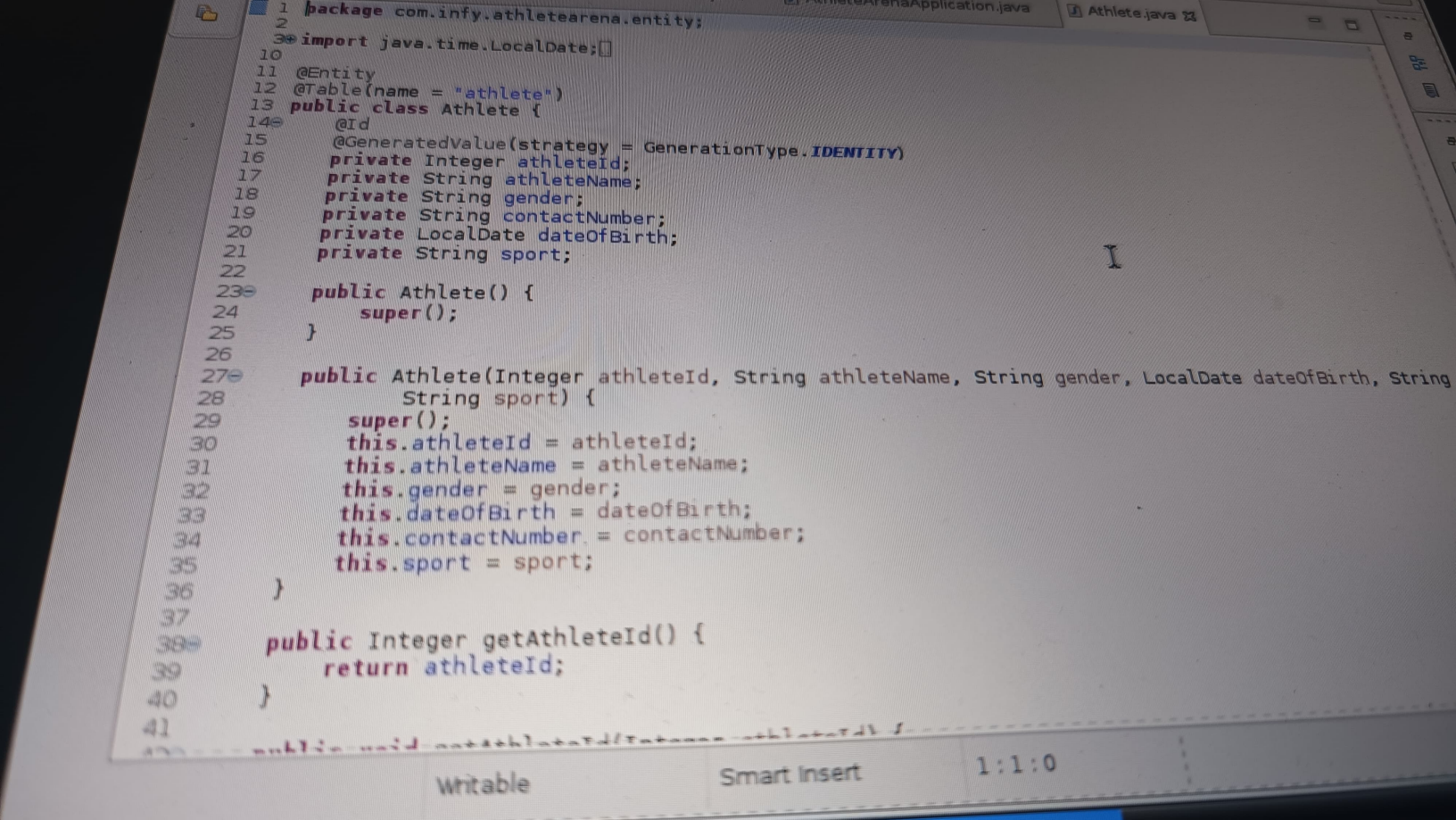

AthleteArenaApplication.java
Athlete.java
1 package com.infy.athletearena.entity;
2
3 import java.time.LocalDate;
10
11 @Entity
12 @Table(name = "athlete")
13 public class Athlete {
14 @Id
15 @GeneratedValue(strategy = GenerationType.IDENTITY)
16 private Integer athleteId;
17 private String athleteName;
18 private String gender;
19 private String contactNumber;
20 private LocalDate dateOfBirth;
21 private String sport;
22
23 public Athlete() {
24 super();
25 }
26
27 public Athlete(Integer athleteId, String athleteName, String gender, LocalDate dateOfBirth, String
28 String sport) {
29 super();
30 this.athleteId = athleteId;
31 this.athleteName = athleteName;
32 this.gender = gender;
33 this.dateOfBirth = dateOfBirth;
34 this.contactNumber = contactNumber;
35 this.sport = sport;
36 }
37
38 public Integer getAthleteId() {
39 return athleteId;
40 }
41
Writable
Smart Insert
1 : 1 : 0

```java
import java.time.LocalDate;

public class AthleteDTO {

    private Integer athleteId;
    @NotNull(message = "{athlete.name.notpresent}")
    @Pattern(regexp = "[a-zA-z]+([\\sa-zA-z])*",message = "{athlete.name.invalid}")
    private String athleteName;
    @NotNull(message = "{athlete.gender.notpresent}")
    @Pattern(regexp = "Male|Female|Others", message = "{athlete.gender.invalid}")
    private String gender;
    private LocalDate dateOfBirth;
    @NotNull(message = "{athlete.contactnumber.notpresent}")
    @Pattern(regexp = "[7-9]{1}[0-9]{9}", message = "{athlete.contactnumber.invalid}")
    private String contactNumber;
    @NotNull(message = "{athlete.sport.notpresent}")
    @Pattern(regexp = "[A-Za-z0-9\s]+", message = "{athlete.sport.invalid}")
    private String sport;

    public AthleteDTO() {
        super();
    }

    public AthleteDTO(Integer athleteId, String athleteName, String gender, LocalDate dateOfBirth,
            String contactNumber, String sport) {
        super();
        this.athleteId = athleteId;
        this.athleteName = athleteName;
        this.gender = gender;
        this.dateOfBirth = dateOfBirth;
        this.contactNumber = contactNumber;
        this.sport = sport;
    }
```

Writable | Smart Insert | 17 : 37 : 500

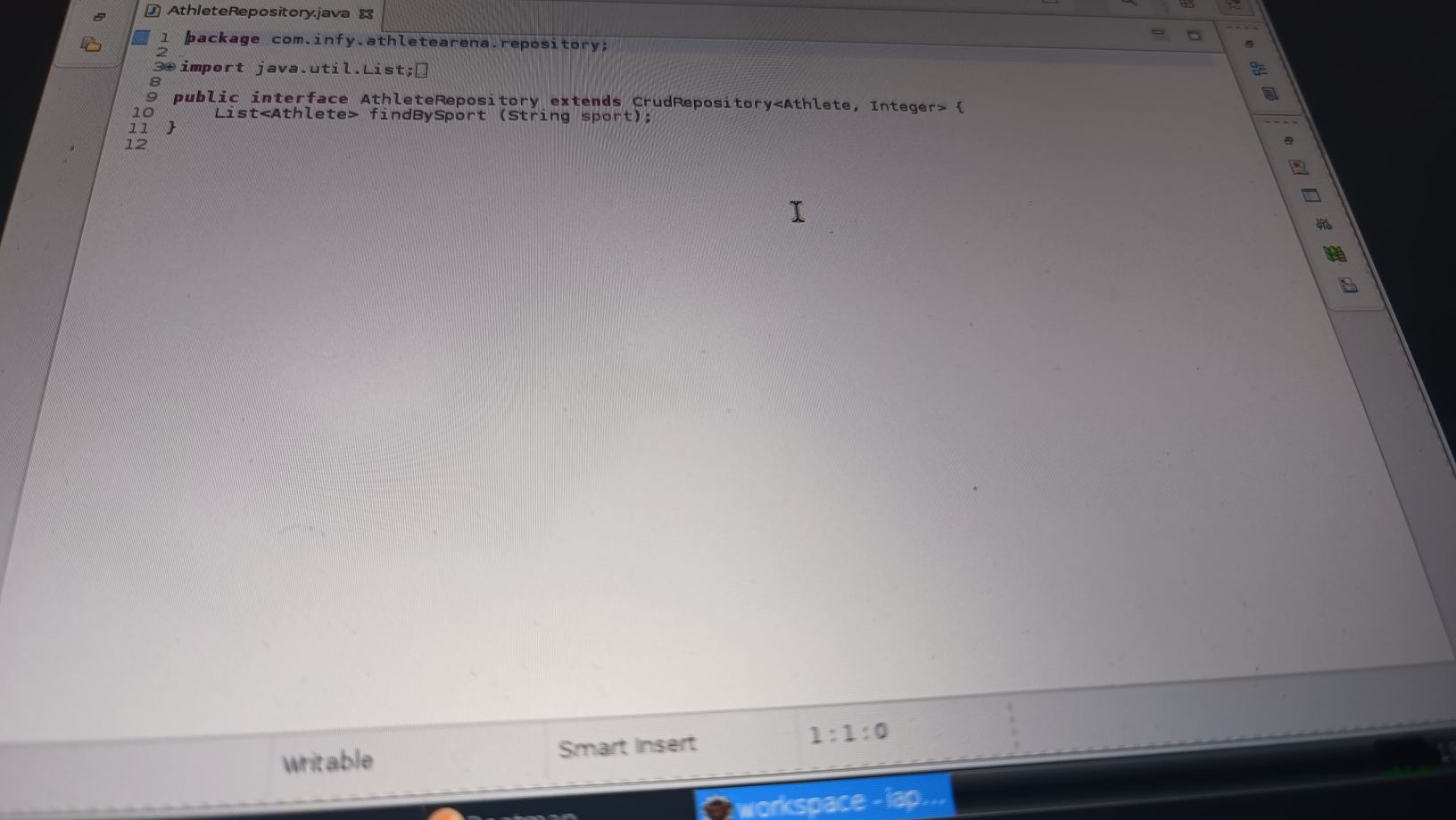
Mozilla Firefox | Postman | workspace - iap...

AthleteValidator.java

```java
package com.infy.athletearena.validator;

import java.time.LocalDate;

public class AthleteValidator {


    public static void validateAthlete(AthleteDTO athleteDTO) throws AthleteRegistrationException{
        if( Boolean.FALSE.equals(isValidDateOfBirth(athleteDTO.getDateOfBirth()))) {
            throw new AthleteRegistrationException("AthleteValidator.INVALID_DOB");
        }
    }

    public static Boolean isValidDateOfBirth(LocalDate dateOfBirth) {
        if(dateOfBirth.isAfter(LocalDate.now()) || dateOfBirth.isBefore(LocalDate.now().minusYears(100))) {
            return false;
        }
        return true;
    }
}

```

Writable | Smart Insert | 1 : 1 : 0

Mozilla Firefox | Postman | workspace - iap... | 18:53


AthleteRepository.java
1 package com.infy.athletearena.repository;
2
3 import java.util.List;
8
9 public interface AthleteRepository extends CrudRepository<Athlete, Integer> {
10 List<Athlete> findBySport (String sport);
11 }
12
Writable
Smart Insert
1 : 1 : 0
workspace - iap...

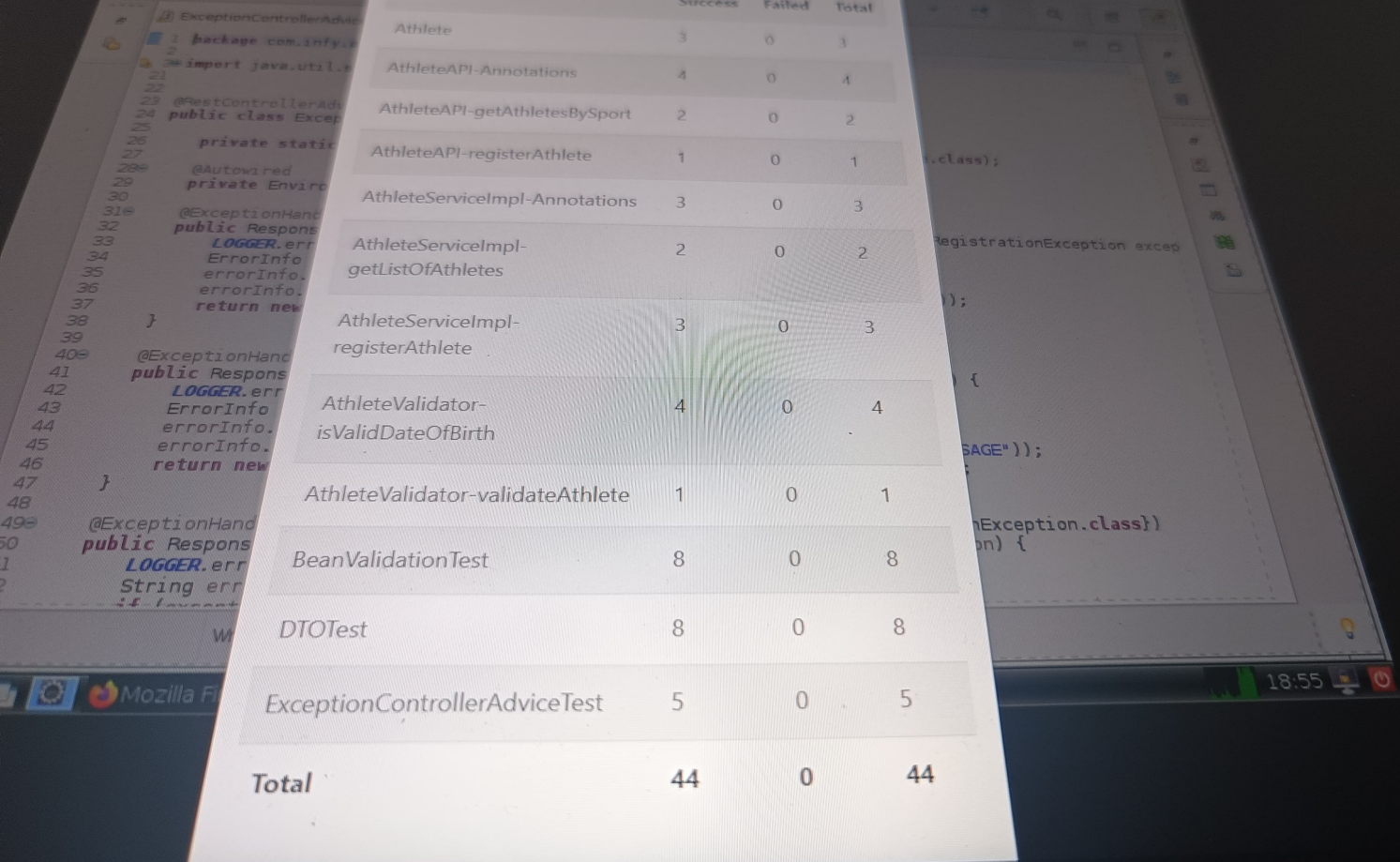
```java
import java.util.ArrayList;

@Service(value = "athleteService")
@Transactional
public class AthleteServiceImpl implements AthleteService{
    @Autowired
    private AthleteRepository athleteRepository;

    @Override
    public List<AthleteDTO> getListOfAthletes(String sport) throws AthleteRegistrationException {
        List<Athlete> athlete=athleteRepository.findBySport(sport);
        if(athlete.isEmpty()) {
            throw new AthleteRegistrationException("AthleteService.ATHLETE_UNAVAILABLE");
        }
        List<AthleteDTO> dto =new ArrayList<AthleteDTO>();
        athlete.stream().forEach(x->dto.add(new AthleteDTO().prepareDTO(x)));
        dto.sort((x1,x2)->x1.getAthleteName().compareTo(x2.getAthleteName()));
        return dto;
    }

    @Override
    public AthleteDTO registerAthlete(AthleteDTO athleteDTO) throws AthleteRegistrationException {
        AthleteValidator.validateAthlete(athleteDTO);
        Athlete a=AthleteDTO.prepareEntity(athleteDTO);
        a=athleteRepository.save(a);
        athleteDTO.setAthleteId(a.getAthleteId());
        return athleteDTO;
    }

}
```

```java
package com.infy.athletesarena.api;
import java.util.List;
@RestController
@RequestMapping(value = "api")
@Validated
public class AthleteAPI {
@Autowired
    private AthleteService athleteService;
    @GetMapping(value = "/athletes/{sport}")
    public ResponseEntity<List<AthleteDTO>> getAthletesBySport(
            @PathVariable @Pattern(regexp = "[A-Za-z0-9\\s]+",
            message = "{athlete.sport.invalid}") String sport)
            throws AthleteRegistrationException {
        List<AthleteDTO> aDTO=athleteService.getListOfAthletes(sport);
        return new ResponseEntity<>(aDTO,HttpStatus.OK);
    }

@PostMapping(value = "/athletes")
    public ResponseEntity<AthleteDTO> registerAthlete( AthleteDTO athleteDTO)
            throws AthleteRegistrationException {
    AthleteDTO a=athleteService.registerAthlete(athleteDTO);
        return new ResponseEntity<>(a, HttpStatus.CREATED);
    }
}

```

```java
@RestControllerAdvice
public class ExceptionControllerAdvice {

    private static final Log LOGGER = LogFactory.getLog(ExceptionControllerAdvice.class);

    @Autowired
    private Environment environment;

    @ExceptionHandler(AthleteRegistrationException.class)
    public ResponseEntity<ErrorInfo> athleteRegistrationExceptionHandler(AthleteRegistrationException excep
        LOGGER.error(exception.getMessage(), exception);
        ErrorInfo errorInfo = new ErrorInfo();
        errorInfo.setErrorCode(HttpStatus.BAD_REQUEST.value());
        errorInfo.setErrorMessage(environment.getProperty(exception.getMessage()));
        return new ResponseEntity<>(errorInfo, HttpStatus.BAD_REQUEST);
    }

    @ExceptionHandler(Exception.class)
    public ResponseEntity<ErrorInfo> generalExceptionHandler(Exception exception) {
        LOGGER.error(exception.getMessage(), exception);
        ErrorInfo errorInfo = new ErrorInfo();
        errorInfo.setErrorCode(HttpStatus.INTERNAL_SERVER_ERROR.value());
        errorInfo.setErrorMessage(environment.getProperty("General.EXCEPTION_MESSAGE"));
        return new ResponseEntity<>(errorInfo, HttpStatus.INTERNAL_SERVER_ERROR);
    }

    @ExceptionHandler({MethodArgumentNotValidException.class, ConstraintViolationException.class})
    public ResponseEntity<ErrorInfo> validatorExceptionHandler(Exception exception) {
        LOGGER.error(exception.getMessage(), exception);
        String errorMsg;
```

Writable | Smart Insert | 1 : 1 : 0

Mozilla Firefox | Postman | workspace - iap...



	Success	Failed	Total
Athlete	3	0	3
AthleteAPI-Annotations	4	0	4
AthleteAPI-getAthletesBySport	2	0	2
AthleteAPI-registerAthlete	1	0	1
AthleteServiceImpl-Annotations	3	0	3
AthleteServiceImpl-getListOfAthletes	2	0	2
AthleteServiceImpl-registerAthlete	3	0	3
AthleteValidator-isValidDateOfBirth	4	0	4
AthleteValidator-validateAthlete	1	0	1
BeanValidationTest	8	0	8
DTOTest	8	0	8
ExceptionControllerAdviceTest	5	0	5
Total	**44**	**0**	**44**