# Contexte de session

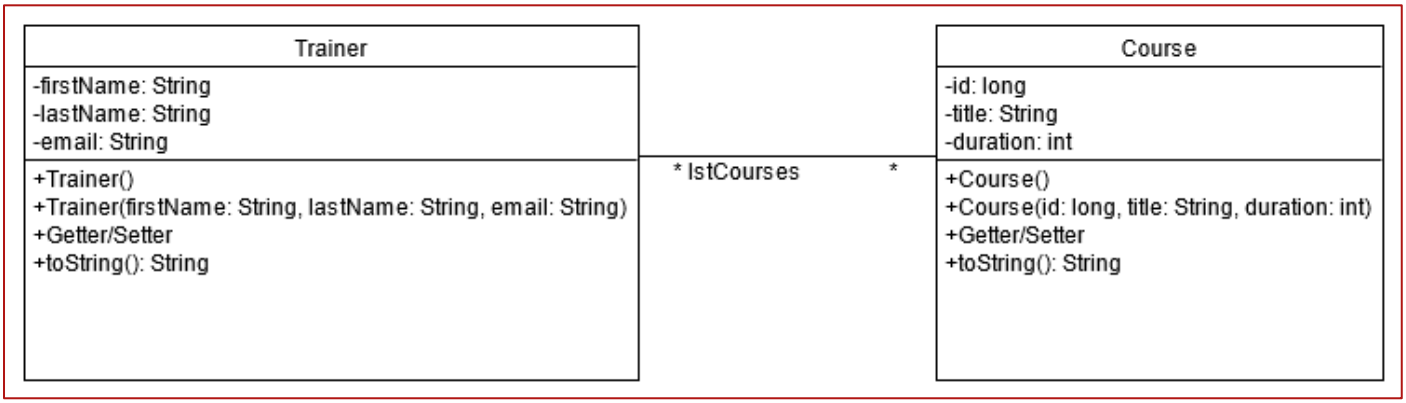
## Démonstration 6 du module 4

L'objectif de cette démonstration est de mettre en avant la manipulation du contexte de session par Spring.

# Contexte

## Compléter le projet précédent

- Pour le moment, l'application ne gérait que les formateurs.
- Maintenant, il est possible d'associer à un formateur ses cours qu'il dispense.
- Voici le diagramme des BO :




- La liste des cours de l'ENI n'évoluent pas tous les jours.
- Si nous devons associer un cours à un formateur, il serait intéressant que cette liste soit au minimum en session pour éviter de la récupérer continuellement.

# Déroulement

## Compléter les BO

- Création de la classe Course :

```
package fr.eni.demo.bo;

public class Course {
	private long id;
	private String title;
	private int duration;

	public Course() {
	}

	public Course(long id, String title, int duration) {
		this.id = id;
		this.title = title;
```

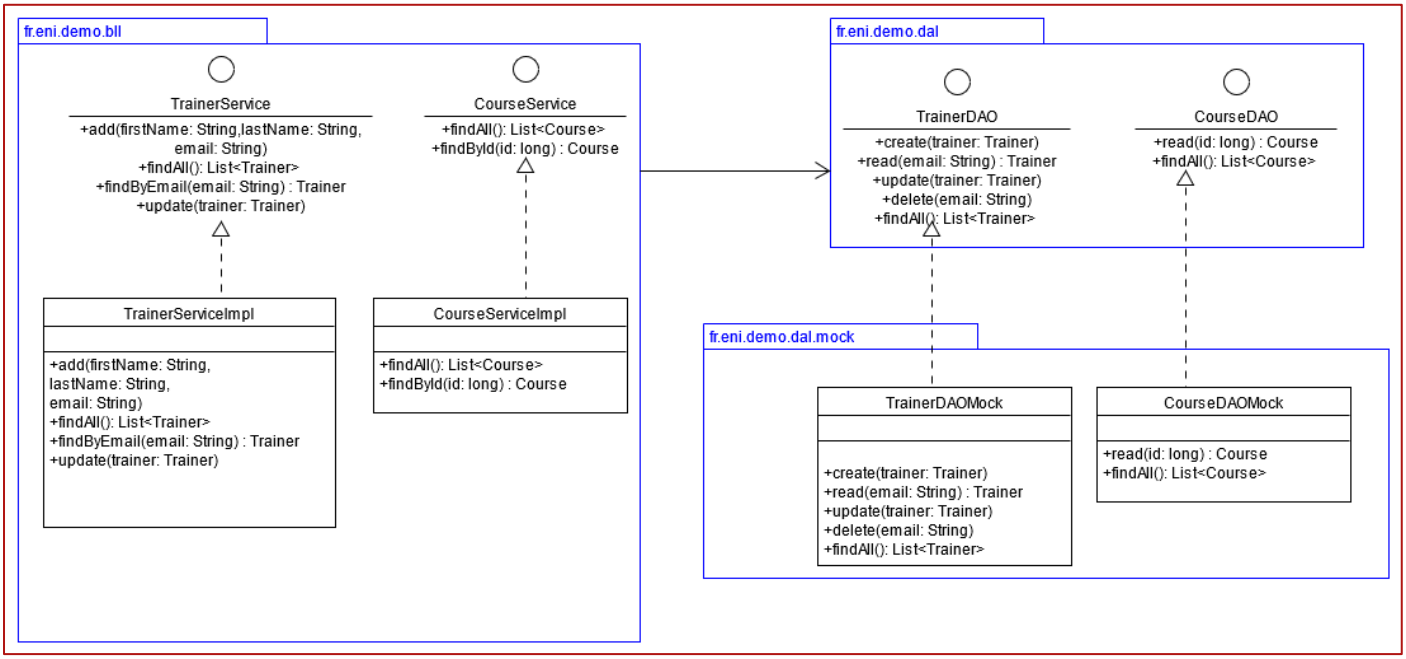
```
		this.duration = duration;
	}

	// + Getter/Setter


	@Override
	public String toString() {
		StringBuilder builder = new StringBuilder();
		builder.append("course [id=");
		builder.append(id);
		builder.append(", title=");
		builder.append(title);
		builder.append(", duration=");
		builder.append(duration);
		builder.append("]");
		return builder.toString();
	}
}
```

- Ajout de l'association dans la classe Trainer :

```
package fr.eni.demo.bo;

import java.util.ArrayList;
import java.util.List;

public class Trainer {
	private String firstName;
	private String lastName;
	private String email;

	//1 formateur peut dispenser plusieurs cours
	private List<Course> lstCourses;

	public Trainer() {
		//initialisation de la liste des cours.
		lstCourses = new ArrayList<Course>();
	}

	public Trainer(String firstName, String lastName, String email) {
		this();//appel du constructeur par défaut
		this.firstName = firstName;
		this.lastName = lastName;
		this.email = email;
	}

	// + Getter/Setter

	@Override
	public String toString() {
		return "Trainer [firstName=" + firstName + ", lastName=" + lastName + ", email=" + email +
"]";
	}
}
```

## Compléter les couches DAL et BLL

- Il est possible de récupérer l'ensemble des cours et rechercher un cours par son identifiant.
- Il est possible de mettre à jour un formateur en ajoutant ses cours associés




**CourseDAO :**

```
package fr.eni.demo.dal;

import java.util.List;

import fr.eni.demo.bo.Course;

public interface CourseDAO {
	Course read(long id);

	List<Course> findAll();
}
```

**CourseDAOMock :**

```
package fr.eni.demo.dal.mock;

import java.util.ArrayList;
import java.util.List;

import org.springframework.stereotype.Repository;

import fr.eni.demo.bo.Course;
import fr.eni.demo.dal.CourseDAO;

@Repository
public class CourseDAOMock implements CourseDAO {
	// Solution temporaire - gestion d'une liste de formateur locale
	private static List<Course> lstCourses;

	public CourseDAOMock() {
		lstCourses = new ArrayList<Course>();
		lstCourses.add(new Course(10, "Algorithmique", 5));
		lstCourses.add(new Course(20, "Initiation à la programmation", 5));
		lstCourses.add(new Course(30, "POO", 10));
		lstCourses.add(new Course(50, "SQL", 5));
```

```
		lstCourses.add(new Course(60, "PL-SQL", 5));
		lstCourses.add(new Course(130, "Web Client", 5));
		lstCourses.add(new Course(140, "Symfony", 10));
		lstCourses.add(new Course(220, "frameworks Java EE", 10));
	}

	@Override
	public Course read(long id) {
		for (Course course : lstCourses) {
			if (course.getId() == id) {
				return course;
			}
		}
		return null;
	}

	@Override
	public List<Course> findAll() {
		return lstCourses;
	}
}
```

**CourseService :**

```
package fr.eni.demo.bll;

import java.util.List;

import fr.eni.demo.bo.Course;

public interface CourseService {
	List<Course> findAll();

	Course findById(long id);
}
```

**CourseServiceImpl :**

```
package fr.eni.demo.bll;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import fr.eni.demo.bo.Course;
import fr.eni.demo.dal.CourseDAO;

@Service
public class CourseServiceImpl implements CourseService {
	private CourseDAO courseDAO;

	@Autowired
	public CourseServiceImpl(CourseDAO courseDAO) {
		this.courseDAO = courseDAO;
	}

	@Override
	public List<Course> findAll() {
		return courseDAO.findAll();
	}

	@Override
	public Course findById(long id) {
		return courseDAO.read(id);
	}
}
```

**Compléter TrainerService :**

```
package fr.eni.demo.bll;

import java.util.List;

import fr.eni.demo.bo.Trainer;

public interface TrainerService {
	…

	void update(Trainer trainer);
}
```

**Compléter TrainerServiceImpl :**

```
package fr.eni.demo.bll;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import fr.eni.demo.bo.Trainer;
import fr.eni.demo.dal.TrainerDAO;

@Service
public class TrainerServiceImpl implements TrainerService {
	private TrainerDAO trainerDAO;

	@Autowired
	public TrainerServiceImpl(TrainerDAO trainerDAO) {
		this.trainerDAO = trainerDAO;
	}

	…

	@Override
	public void update(Trainer trainer) {
		trainerDAO.update(trainer);
	}
}
```

## Gestion de la liste des cours de l'ENI par @SessionAttributes

- La liste des cours de l'ENI est envoyée en session dès l'affichage, la création ou la mise à jour d'un formateur.
- Modification du contrôleur : TrainerController
  - Ajout de @SessionAttributes
  - Injection du service des cours.
  - Ajout d'un attribut en session et de sa méthode associée pour gérer tous les cours de l'ENI.
  - Appel de cette méthode dans detailTrainer
  - Création d'une méthode pour gérer l'ajout d'un cours au formateur

```
package fr.eni.demo.mmi.controller;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.*;

import fr.eni.demo.bll.*;
import fr.eni.demo.bo.*;

@Controller
@RequestMapping("/trainers")
//Mise en session de la liste des cours
@SessionAttributes({"allCourses"})
public class TrainerController {
	private TrainerService trainerService;
	private CourseService courseService;

	@Autowired
	// Injection des services
	public TrainerController(TrainerService trainerService, CourseService courseService) {
		this.trainerService = trainerService;
		this.courseService = courseService;

	}
	…
	@GetMapping("/detail")
	public String detailTrainer(
			@RequestParam(name = "email", required = false,
			 defaultValue = "coach@campus-eni.fr") String emailTrainer, Model model) {
		System.out.println("Le paramètre - " + emailTrainer);
		Trainer trainer = trainerService.findByEmail(emailTrainer);
		// Ajout de l'instance dans le modèle
		model.addAttribute("trainer", trainer);

		//Appel de la méthode de chargement des cours en session
		loadCourses(model);
		return "view-trainer-form";
	}
	…
	// Méthode pour charger la liste des cours en session
	private void loadCourses(Model model) {
		// Vérification que l'attribut n'est pas déjà en session
		Object sessionAttribute = model.getAttribute("allCourses");
		if (sessionAttribute == null) {
			System.out.println("Chargement en session de tous les cours");
			model.addAttribute("allCourses", courseService.findAll());
		}
	}

	//Ajout d'un cours au formateur courant
	@PostMapping("/courses")
	public String addCourse(@RequestParam(required = true) String email,
			@RequestParam(name="addCourse", required = true) String id) {
		Trainer trainer = trainerService.findByEmail(email);
		long idCourse = Long.parseLong(id);
		Course c = courseService.findById(idCourse);
		trainer.getLstCourses().add(c);
		trainerService.update(trainer);
		return "redirect:/trainers/detail?email="+email;
	}
}
```

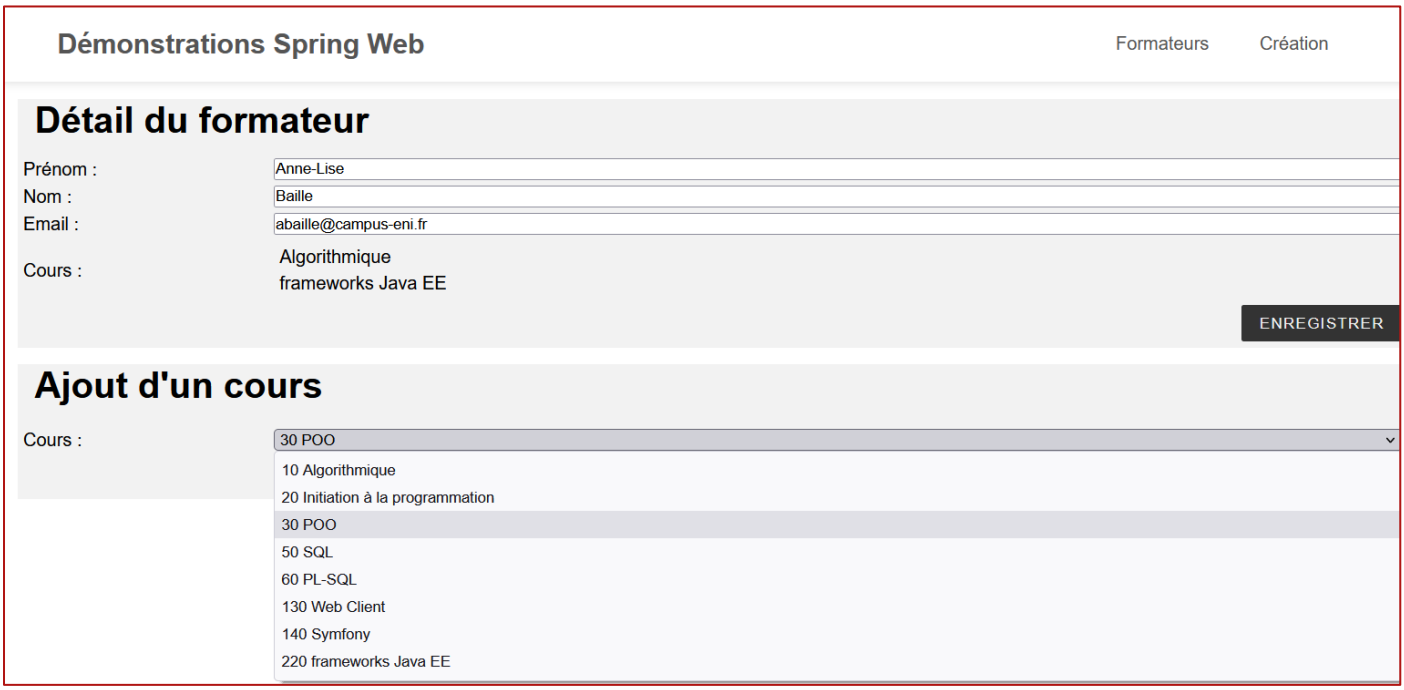
- Modifier l'affichage du détail des formateurs

- Intégrer la liste des cours du formateur
- Ajout d'un formulaire pour ajouter un nouveau cours au formateur, en sélectionnant parmi ceux en session

```
<!DOCTYPE html>
<!-- Ajout du moteur de template Thymeleaf -->
<html xmlns:th="http://www.thymeleaf.org">
<head>
<meta charset="UTF-8">
<title>Demo Spring Web</title>
<link rel="stylesheet" href="/css/general.css">
<link rel="stylesheet" href="/css/demo-form.css">
<link rel="icon" href="/images/logoENI.png">
</head>
<body>
<header>
	<h1 id="nav-title">
		<a href="/">Démonstrations Spring Web</a>
	</h1>
	<nav>
		<ul>
			<li><a href="/trainers">Formateurs</a></li>
			<li><a href="/trainers/create">Création</a></li>
		</ul>
	</nav>
</header>
<main>
	<form action="/trainers" method="post">
		<h1>Détail du formateur</h1>
		<ul class="flex-outer">
			<li><label for="inputFirstN">Prénom : </label>
				<input type="text" name="firstName" id="inputFirstN" required
				data-th-value="${trainer.firstName}" />
			</li>
			<li><label for="inputLastN">Nom : </label>
				<input type="text" name="lastName" id="inputLastN" required
				data-th-value="${trainer.lastName}" />
			</li>
			<li><label for="inputEmail">Email : </label>
				<input type="text" name="email" id="inputEmail" required
				data-th-value="${trainer.email}" />
			</li>
			<!-- Ajout liste des cours d'un formateur -->
			<li><label for="tableCourses">Cours : </label>
				<ul >
					<li data-th-each="current : ${trainer.lstCourses}" data-th-
text="${current.title}"></li>
				</ul>
			</li>
			<li>
				<button type="submit">Enregistrer</button>
			</li>
		</ul>
	</form>
	<!-- Formulaire pour l'ajout d'un cours au formateur -->
	<form action="/trainers/courses" method="post">
		<h1>Ajout d'un cours</h1>
		<ul class="flex-outer">
			<li><input type="hidden" name="email" id="inputEmail" required
				data-th-value="${trainer.email}"/>
			</li>
			<li><label for="addCourse">Cours : </label>
				<select  name="addCourse">
					<option data-th-each="course : ${allCourses}"  data-th-
text="${course.id} + ' ' + ${course.title}" data-th-value="${course.id}"></option>
```

```
                        </select>
                    </li>
                    <li>
                        <button type="submit">Enregistrer</button>
                    </li>
                </ul>
            </form>
</main>
</body>
</html>
```

- Résultats :




- La liste des cours est toujours présent en session
  - Il est possible de revenir sur la vue et d'ajouter autant de cours que nécessaire au formateur sélectionné.
  - Sans qu'il y ait rechargement de cette liste.
- Voici les traces de la console pour le confirmer :

```
…
Le paramètre - abaille@campus-eni.fr
Chargement en session de tous les cours
Le paramètre - abaille@campus-eni.fr
Le paramètre - abaille@campus-eni.fr
```

  - L'ajout de l'attribut n'est fait que la première fois qu'il a y le détail d'un formateur.