**package** com.cts.controller;

**import** com.cts.entity.User;

**import** com.cts.service.UserService;

**import** org.springframework.beans.factory.annotation.Autowired;

**import** org.springframework.hateoas.\*;

**import** org.springframework.http.ResponseEntity;

**import** org.springframework.web.bind.annotation.\*;

**import** java.util.List;

**import** java.util.stream.Collectors;

**import** **static** org.springframework.hateoas.server.mvc.WebMvcLinkBuilder.\*;

@RestController

@RequestMapping("/users")

**public** **class** UserController {

@Autowired

**private** UserService userService;

@GetMapping("/all")

**public** ResponseEntity<CollectionModel<EntityModel<User>>> getUsers() {

List<User> users = userService.getAllUsers();

List<EntityModel<User>> userModels = users.stream()

.map(user -> EntityModel.*of*(user,

*linkTo*(*methodOn*(UserController.**class**).getUserById(user.getId())).withSelfRel(),

*linkTo*(*methodOn*(UserController.**class**).getUsers()).withRel("all-users")))

.collect(Collectors.*toList*());

CollectionModel<EntityModel<User>> collectionModel = CollectionModel.*of*(userModels,

*linkTo*(*methodOn*(UserController.**class**).getUsers()).withSelfRel());

**return** ResponseEntity.*ok*(collectionModel);

}

@GetMapping("/{id}")

**public** ResponseEntity<EntityModel<User>> getUserById(@PathVariable Long id) {

User user = userService.getUserById(id);

EntityModel<User> userModel = EntityModel.*of*(user,

*linkTo*(*methodOn*(UserController.**class**).getUserById(id)).withSelfRel(),

*linkTo*(*methodOn*(UserController.**class**).getUsers()).withRel("all-users"));

**return** ResponseEntity.*ok*(userModel);

}

@PostMapping("/save")

**public** ResponseEntity<EntityModel<User>> createUser(@RequestBody User user) {

User savedUser = userService.saveUser(user);

EntityModel<User> userModel = EntityModel.*of*(savedUser,

*linkTo*(*methodOn*(UserController.**class**).getUserById(savedUser.getId())).withSelfRel(),

*linkTo*(*methodOn*(UserController.**class**).getUsers()).withRel("all-users"));

**return** ResponseEntity

.*created*(*linkTo*(*methodOn*(UserController.**class**).getUserById(savedUser.getId())).toUri())

.body(userModel);

}

@PutMapping("/update/{id}")

**public** ResponseEntity<EntityModel<User>> updateUser(@PathVariable Long id, @RequestBody User updatedUser) {

User user = userService.updateUser(id, updatedUser);

EntityModel<User> userModel = EntityModel.*of*(user,

*linkTo*(*methodOn*(UserController.**class**).getUserById(user.getId())).withSelfRel(),

*linkTo*(*methodOn*(UserController.**class**).getUsers()).withRel("all-users"));

**return** ResponseEntity.*ok*(userModel);

}

@DeleteMapping("/{id}")

**public** ResponseEntity<?> deleteUserById(@PathVariable Long id) {

userService.deleteUserById(id);

**return** ResponseEntity.*noContent*().build();

}

}