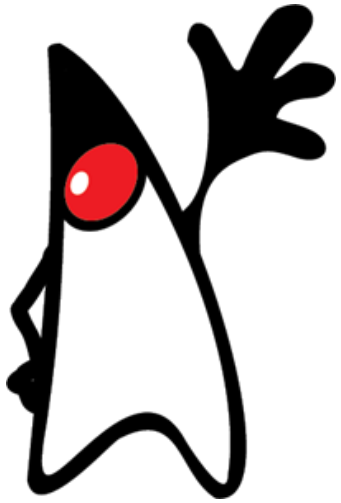
Debugging
part 2



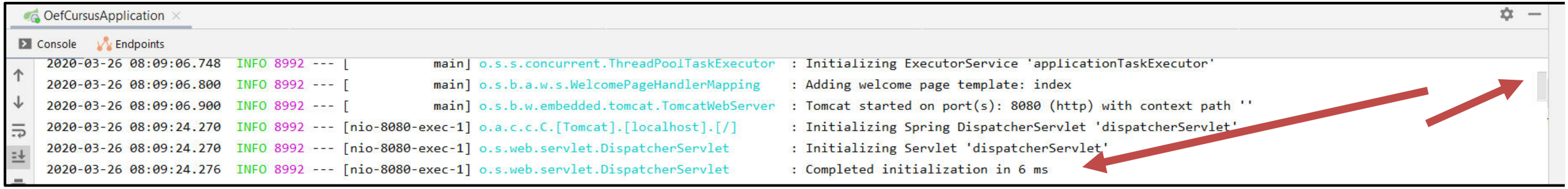
Recap

- Testing and Debugging = locating and fixing errors
- 5 types:
  - opening and installation errors:
    - already visible during the opening of a project
  - compilation errors:
    - during programming
  - testing errors:
    - tests do not run
    - tests do not succeed
  - start-up errors:
    - when you want to run your programme, but it does not start
  - runtime errors: during the execution of your program:
    - your programme runs and crashes at the start or during execution
  - logical errors: the programme does not give the expected / logical result
- Debugger
  - can help you detect runtime errors and logical errors
    - **not** for installation errors, compilation errors and start-up errors

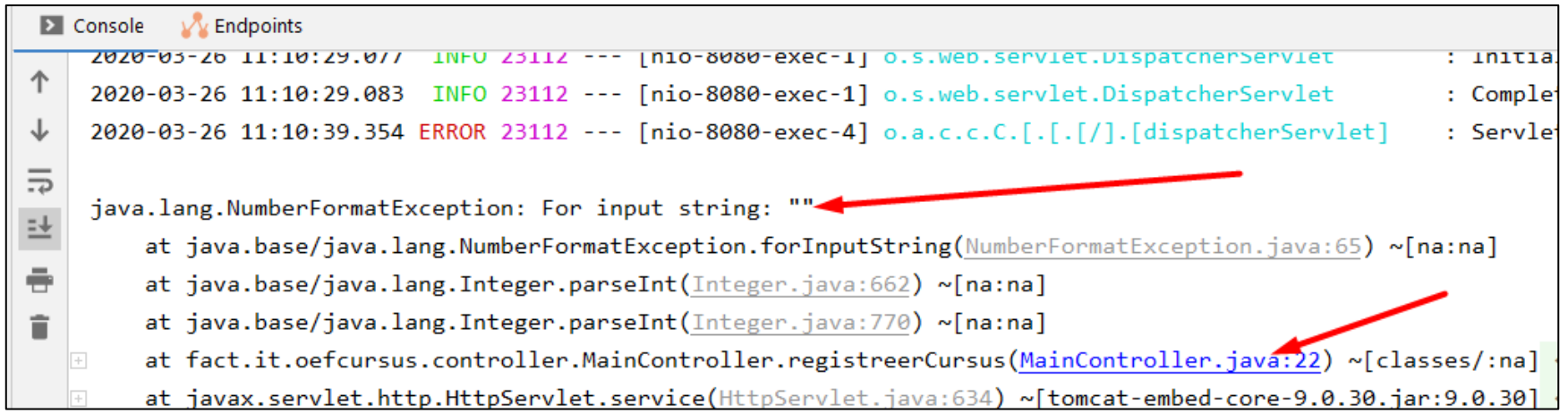
+ Thymeleaf-specific errors!

Runtime errors - Finding the cause

- ALWAYS go to IntelliJ and look in your output window below:



- Scroll to almost the top of this window until this text appear, the lines below are all errors that occur as a result of that cause:



Runtime errors - HTTP Status 404

- The problem usually lies in the fact that your Http request does not match the RequestMapping value in the Controller

```
@RequestMapping("/nieuwe-reservati")
public String nieuweReservatie(Model model) {
```

- Either you change your href attribute in the <a> tag or the action attribute in the <form> tag

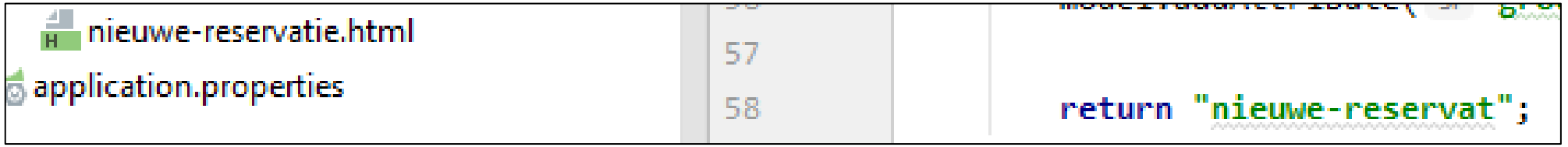
```
<a href="/n
```

```
<form action="/n
```

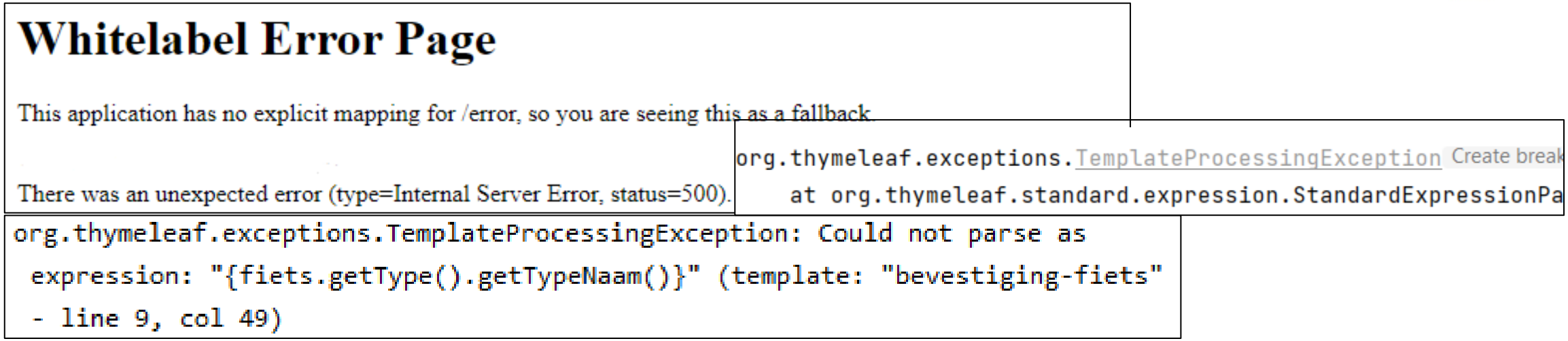
Runtime errors - HTTP 500 – Resolving Template


```
org.thymeleaf.exceptions.TemplateInputException Create breakpoint : Error resolving templa
```

- In the Controller, you refer via return "xxx" to an html page that does not exist.
- Check that your specified HTML page exists, and its name matches.



Thymeleaf Errors – HTTP 500 – ProcessingEx.



- You forgot to put a $ or an @ before the { }. Add the necessary character.

```
<span th:text="{fiets.getType().getTypeNaam()}"/> in gebrui
```

```
<form th:action="{/bevestiging-fiets}"
```

Thymeleaf Errors - Property cannot be found


```
exception is org.thymeleaf.exceptions.TemplateInputException: An error happened
during template parsing (template: "class path resource
[templates/bevestiging-fiets.html]")] with root cause

org.springframework.expression.spel.SpelEvaluationException: EL1008E: Property or
 field 'getTypeNaam' cannot be found on object of type 'fact.it.voorbeeldfietsen
.model.FietsType' - maybe not public or not valid?
```

- You forgot ( ) after a method:

```
<span th:text="${fiets.getType().getTypeNaam}"/> in ge
```

- Or… the method is used correctly, but the implementation of the method in the class has a problem that you should check.

Debugging with the IntelliJ Debugger

- A detailed demo of how to use the IntelliJ debugger to debug your program can be seen in this video:

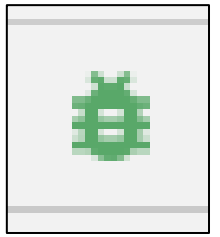
  https://www.youtube.com/watch?v=1bCgzjatcr4

- In the next slides you find a summary of this content…

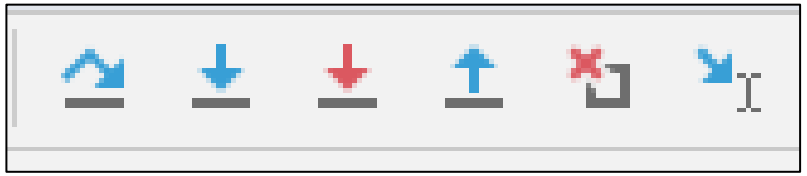
Debugging with the IntelliJ Debugger

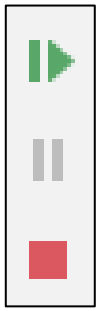
- When you use a debugger, you put red breakpoints in the program. A breakpoint is a place in the program where it will stop.

```
29      @RequestMapping("/bevestiging-fiets")
30      public String bevestigingFiets(Model model, HttpServletRequest request){
31          String grootte = request.getParameter( s: "grootte");
32          String metStang = request.getParameter( s: "metStang");
33          int jaartal = Integer.parseInt(request.getParameter( s: "jaartal"));
```



- Then you can run the program in debug mode by clicking on the bug icon at the top.
- With the controls at the bottom of the Debugger/Console window you can control the execution of your program yourself:



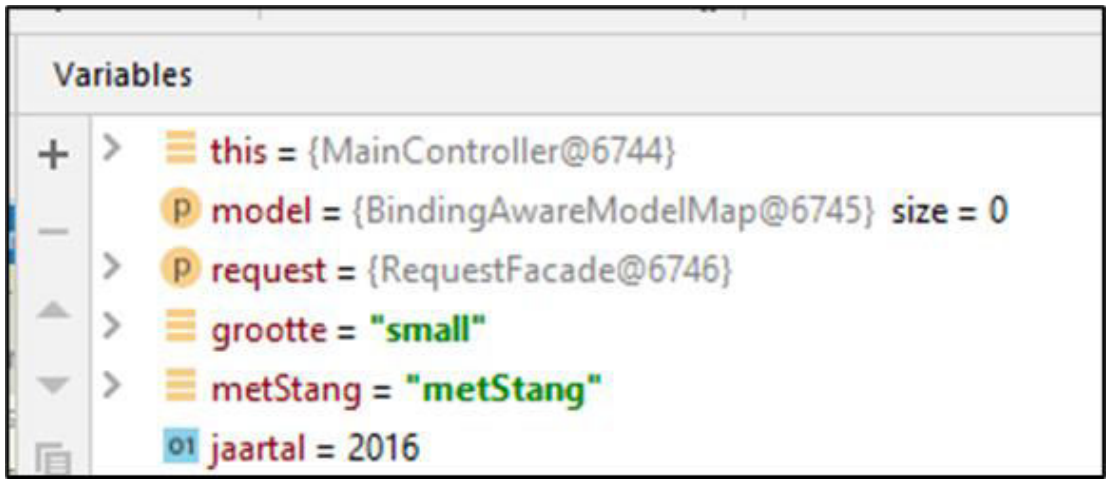


Debugging with the IntelliJ Debugger

- The current contents of the variables are listed at the end as information, and can be accessed by hovering over a variable:

```
public String bevestigingFiets(Model model, HttpServletRequest request){  model:
    String grootte = request.getParameter( s: "grootte");  grootte: "small"
    String metStang = request.getParameter( s: "metStang");  metStang: "metStang"
    int j + "small"  eger.parseInt(request.getParameter( s: "jaartal"));  request
```

- The contents of the variables can also be viewed in the variables tab at the bottom:



Debugging with the IntelliJ Debugger

- A productive way to use the debugger:
  1. Try to deduce from the symptoms of the error where it might be located. This will probably narrow the search down to 2-3 methods.
  2. Place a breakpoint at the beginning and end of these suspicious methods.
  3. Run the program with the debugger. Look at the values of the parameters when the program stops at the beginning of a method. At the end, look at the result returned by the method and the values of the relevant variables. In this way, identify in which method the error is located.
  4. Run the program again with the debugger and stop at the beginning of the method in which the error occurs. Then go through the method line by line until you see a difference between reality and what you expect.
  5. You have (hopefully) found the error!

- This presentation didn’t cover everything… take a look at the document on Canvas as well!

The troubleshooting exercise… Primedebug

- **The application can be used to find all the primes up to a specific value.**
- The **IsPrime method** checks:
  - whether the number is lower than or equal to 1 (in which case it is not a prime number)
  - then whether it's 2 (which is the only even prime number)
  - and whether it's even (in which case it's not a prime number)
- If these checks are indecisive, number is an odd number greater than 2. In this case, we simply loop over all the numbers between 3 and the square root of number (which is sufficient to determine if it is a prime).
- Problems:
  - Solve Thymeleaf and HTTP errors at runtime
  - Print out all the primes up to 100. Unfortunately, the code doesn't seem to work correctly → it reports 9, 15, 25, etc. as prime numbers, even though they clearly are not.
  - However there does not seem to be an obvious problem with the code. You must debug the application to heck the IsPrime method and its loops.

Recap Tips & tricks

- Googling "blindly" and doing what is suggested often causes more errors than before
- Only start "Googling" AFTER
  - you have read the error carefully and do not find the solution on that basis
  - you have not found the solution on the basis of the above
- "Smart" googling =
  - give **context** to the error:
    - IntelliJ
    - Java
    - Thymeleaf
    - MySQL
    - …
  - copy error message literally
    - links at the top
    - stackoverflow

Recap Tips & tricks

- consulting fellow students / social media / ...
  - compared to the previous steps
    - increased risk of errors
    - less/no learning effect: this source cannot/should not be consulted during evaluation moments => risk of being judged as “fraud”
- official teaching channels (Canvas, Teams, Outlook Mail):
  - 100% sure of correct information
  - learning effect is greater