The PrimeCheckApp Case Study: Structure & Functionality of the Client

Douglas C. Schmidt

d.schmidt@vanderbilt.edu

www.dre.vanderbilt.edu/~schmidt

Professor of Computer Science

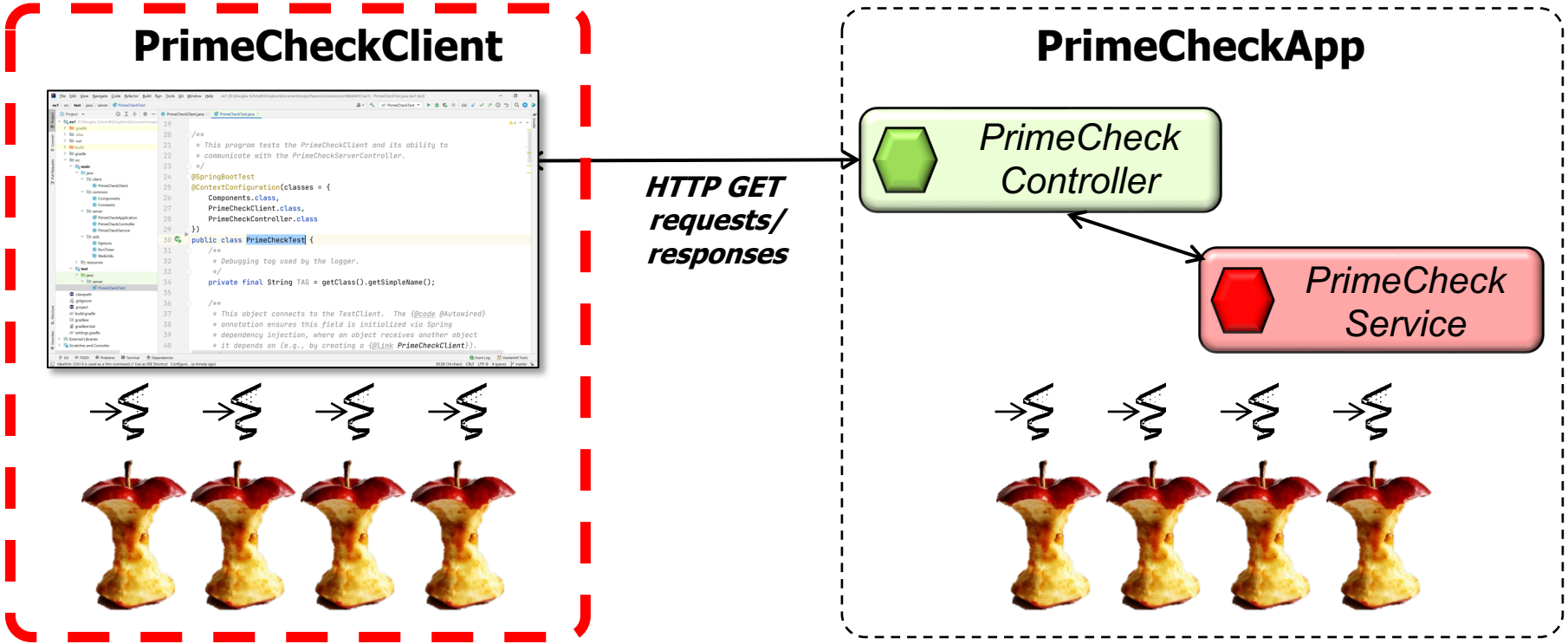
Institute for Software Integrated Systems

Vanderbilt University

Nashville, Tennessee, USA

Learning Objectives in this Part of the Lesson

- Understand the structure & functionality of the PrimeCheckClient that sends/ receives HTTP GET requests/responses to/from the PrimeCheckApp server



See github.com/douglascraigschmidt/LiveLessons/tree/master/WebMVC/ex1

The Structure & Functionality of PrimeCheckClient

The Structure & Functionality of PrimeCheckClient

- The PrimeCheckClient class performs synchronous remote method invocations on the PrimeCheckController to determine the primality of large integers

```
@Component
public class PrimeCheckClient
  @Autowired RestTemplate mRestTemplate;

  public List<Integer> testIndividualCalls
    (List<Integer> primeCandidates,
     boolean parallel) {...}

  public List<Integer>
  testListCall
    (List<Integer> primeCandidates,
     boolean parallel) {...}
}
```

See WebMVC/ex1/src/main/java/client/PrimeCheckClient.java

The Structure & Functionality of PrimeCheckClient

- The PrimeCheckClient class performs synchronous remote method invocations on the PrimeCheckController to determine the primality of large integers

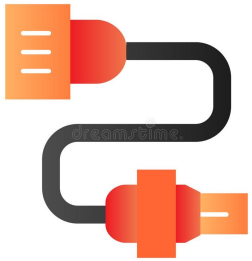
This annotation enables the auto-detection & wiring of dependent implementation classes via classpath scanning

```
@Component
public class PrimeCheckClient
   @Autowired RestTemplate mRestTemplate;

   public List<Integer> testIndividualCalls
      (List<Integer> primeCandidates,
       boolean parallel) {...}

   public List<Integer>
   testListCall
      (List<Integer> primeCandidates,
       boolean parallel) {...}
}
```

See www.baeldung.com/spring-component-repository-service

The Structure & Functionality of PrimeCheckClient

- The PrimeCheckClient class performs synchronous remote method invocations on the PrimeCheckController to determine the primality of large integers



This field is auto-wired by Spring's dependency injection framework

```
@Component
public class PrimeCheckClient
   @Autowired RestTemplate mRestTemplate;

   public List<Integer> testIndividualCalls
      (List<Integer> primeCandidates,
       boolean parallel) {...}

   public List<Integer>
   testListCall
      (List<Integer> primeCandidates,
       boolean parallel) {...}
}
```

See www.baeldung.com/spring-autowire

The Structure & Functionality of PrimeCheckClient

- The PrimeCheckClient class performs synchronous remote method invocations on the PrimeCheckController to determine the primality of large integers

```
@Component
public class PrimeCheckClient
  @Autowired RestTemplate mRestTemplate;

  public List<Integer> testIndividualCalls
    (List<Integer> primeCandidates,
     boolean parallel) {...}

  public List<Integer>
  testListCall
    (List<Integer> primeCandidates,
     boolean parallel) {...}
}
```

Sends individual HTTP GET requests to the server

The Structure & Functionality of PrimeCheckClient

- The PrimeCheckClient class performs synchronous remote method invocations on the PrimeCheckController to determine the primality of large integers

```
List<Integer> testIndividualCalls(List<Integer> primeCandidates,
                                  boolean parallel) {
  var stream = primeCandidates
    .stream();

  if (parallel) stream.parallel();

  return stream.map(primeCandidate -> WebUtils
                    .makeGetRequest
                      (mRestTemplate,
                       makeCheckIfPrimeUrl(primeCandidate),
                       Integer.class))
               .collect(toList());
```

This implementation conditionally enables a parallel or sequential stream using the 'parallel' param

Any requested parallelism is performed by the client rather than by the server

The Structure & Functionality of PrimeCheckClient

- The PrimeCheckClient class performs synchronous remote method invocations on the PrimeCheckController to determine the primality of large integers

```
List<Integer> testIndividualCalls(List<Integer> primeCandidates,
                                  boolean parallel) {
  var stream = primeCandidates
    .stream();

  if (parallel) stream.parallel();

  return stream.map(primeCandidate -> WebUtils
                    .makeGetRequest
                      (mRestTemplate,
                       makeCheckIfPrimeUrl(primeCandidate),
                       Integer.class))
               .collect(toList());
```

Creates a URL that can be passed to an HTTP GET request to determine if an Integer is prime

e.g., http://localhost:8081/checkIfPrime?primeCandidate=2147483515

The Structure & Functionality of PrimeCheckClient

- The PrimeCheckClient class performs synchronous remote method invocations on the PrimeCheckController to determine the primality of large integers

```
List<Integer> testIndividualCalls(List<Integer> primeCandidates,
                                  boolean parallel) {
  var stream = primeCandidates
     .stream();

  if (parallel) stream.parallel();

  return stream.map(primeCandidate -> WebUtils
                    .makeGetRequest
                      (mRestTemplate,
                       makeCheckIfPrimeUrl(primeCandidate),
                       Integer.class))
               .collect(toList());
```

Make an HTTP GET call to the server at the designed URL for each primeCandidate

This helper method shields the client from low-level HTTP programming details

The Structure & Functionality of PrimeCheckClient

- The PrimeCheckClient class performs synchronous remote method invocations on the PrimeCheckController to determine the primality of large integers

Sends a List of Integer objects in one HTTP GET request to the server

```
@Component
public class PrimeCheckClient
  @Autowired RestTemplate mRestTemplate;

  public List<Integer> testIndividualCalls
    (List<Integer> primeCandidates,
     boolean parallel) {...}

  public List<Integer>
  testListCall
    (List<Integer> primeCandidates,
     boolean parallel) {...}
}
```

The Structure & Functionality of PrimeCheckClient

- The PrimeCheckClient class performs synchronous remote method invocations on the PrimeCheckController to determine the primality of large integers

```
List<Integer> testListCalls(List<Integer> primeCandidates,
                            boolean parallel) {

  var getRequestUrl = makeCheckIfPrimeListUrl
    (WebUtils
     .list2String(primeCandidates),
     parallel);

  return WebUtils
    .makeGetRequestList(mRestTemplate,
                        getRequestUrl,
                        Integer[].class);
```

Any requested parallelism is performed by the server rather than by the client

The Structure & Functionality of PrimeCheckClient

- The PrimeCheckClient class performs synchronous remote method invocations on the PrimeCheckController to determine the primality of large integers

```
List<Integer> testListCalls(List<Integer> primeCandidates,
                            boolean parallel) {

  var getRequestUrl = makeCheckIfPrimeListUrl
    (WebUtils
     .list2String(primeCandidates),
     parallel);

  return WebUtils
    .makeGetRequestList(mRestTemplate,
                        getRequestUrl,
                        Integer[].class);
```

Convert the List of Integer objects into a String of comma-separated integers encodings

e.g., "218315,42673259,212438568,147483,5489341,81931857,..."

The Structure & Functionality of PrimeCheckClient

- The PrimeCheckClient class performs synchronous remote method invocations on the PrimeCheckController to determine the primality of large integers

```
List<Integer> testListCalls(List<Integer> primeCandidates,
                            boolean parallel) {

  var getRequestUrl = makeCheckIfPrimeListUrl
     (WebUtils
      .list2String(primeCandidates),
      parallel);

  return WebUtils
     .makeGetRequestList(mRestTemplate,
                         getRequestUrl,
                         Integer[].class);
```

Creates a URL passed via an HTTP GET request to determine the primality of a String of comma-separated integers

e.g., http://localhost:8081/checkIfPrimeList?primeCandidates=218315,147483,...&parallel=true

The Structure & Functionality of PrimeCheckClient

- The PrimeCheckClient class performs synchronous remote method invocations on the PrimeCheckController to determine the primality of large integers

```
List<Integer> testListCalls(List<Integer> primeCandidates,
                            boolean parallel) {

  var getRequestUrl = makeCheckIfPrimeListUrl
    (WebUtils
     .list2String(primeCandidates),
     parallel);

  return WebUtils
    .makeGetRequestList(mRestTemplate,
                        getRequestUrl,
                        Integer[].class);
```

Make one HTTP GET call to the server that contains all encoded primeCandidates

This helper method shields the client from low-level HTTP programming details

End of the PrimeCheck App Case Study: Structure & Functionality of the Client