Architecture of Enterprise Applications 1
Overview of Enterprise Applications

Haopeng Chen

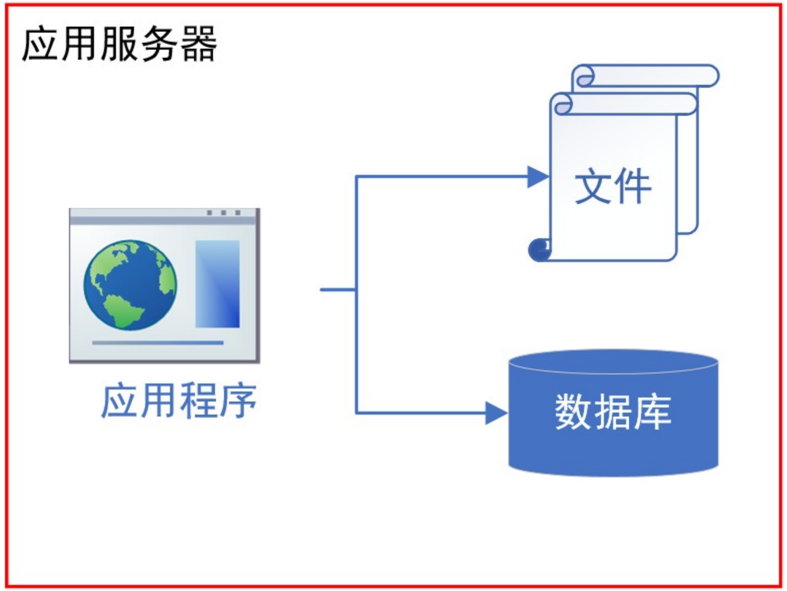
RE*liable,* ***IN****telligent and* ***S****calable Systems Group (****REINS****)*
Shanghai Jiao Tong University
Shanghai, China
http://reins.se.sjtu.edu.cn/~chenhp
e-mail: chen-hp@sjtu.edu.cn

Contents and Objectives

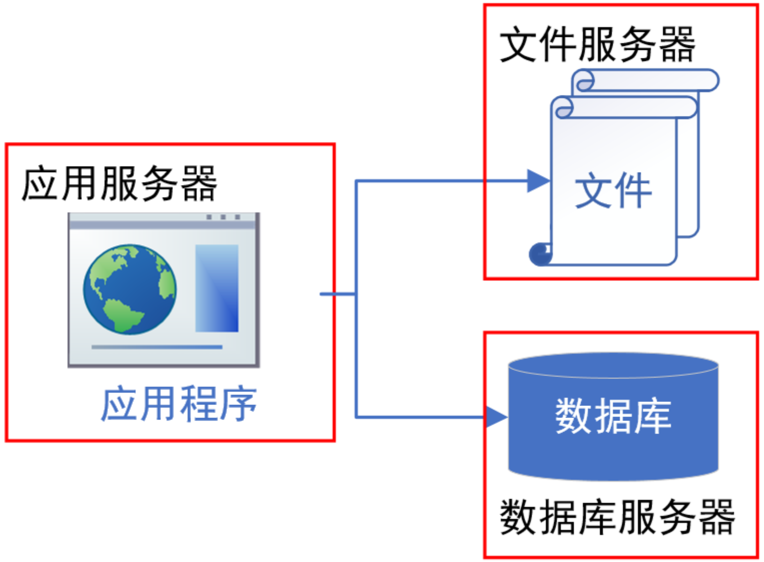
- Contents
  - Architecture
  - Scope

- Objectives
  - 能够根据应用系统的规模，确定适合的技术路线，从单机部署到分布式集群部署，再到微服务架构和云部署
  - 能够根据系统需求，设计并实现由合理的有状态服务与无状态服务构成服务层

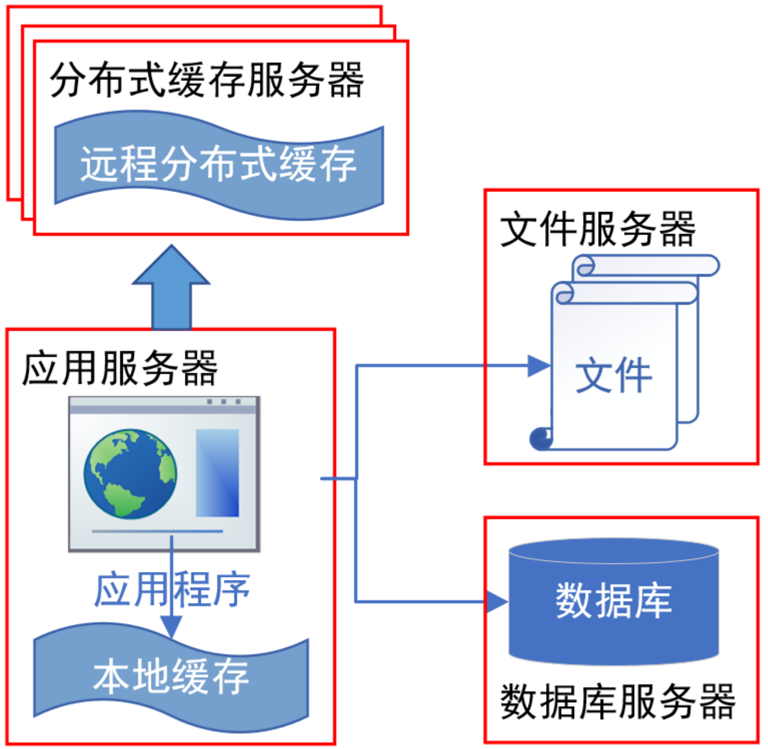
Architecture of Java Web Application



Architecture of Java Web Application

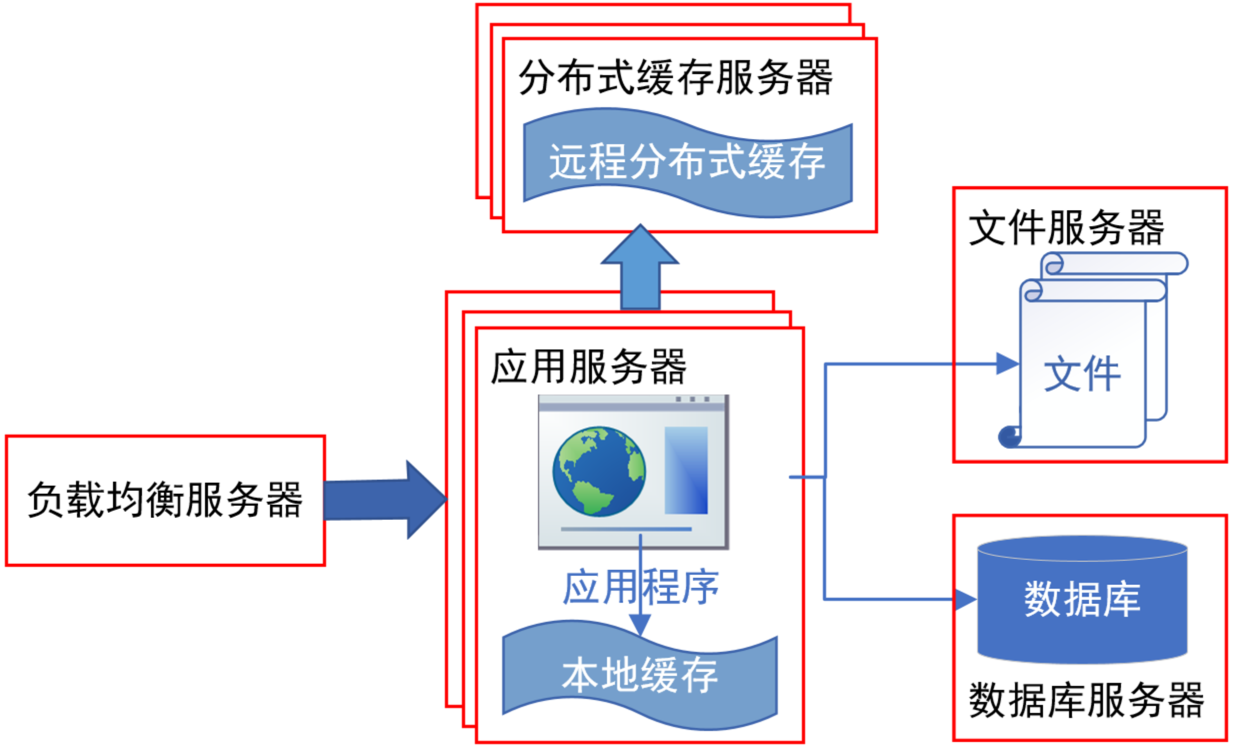


Architecture of Java Web Application



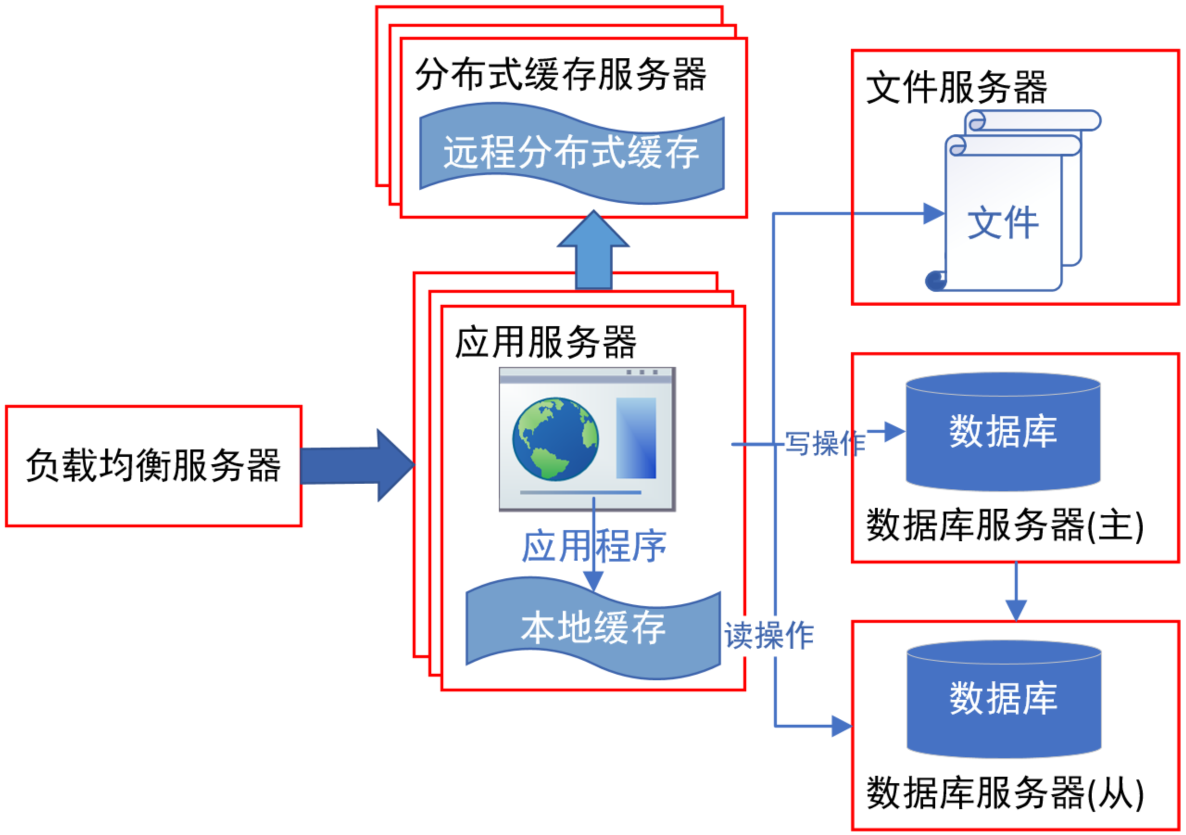
Architecture of Java Web Application

REliable, INtelligent & Scalable Systems



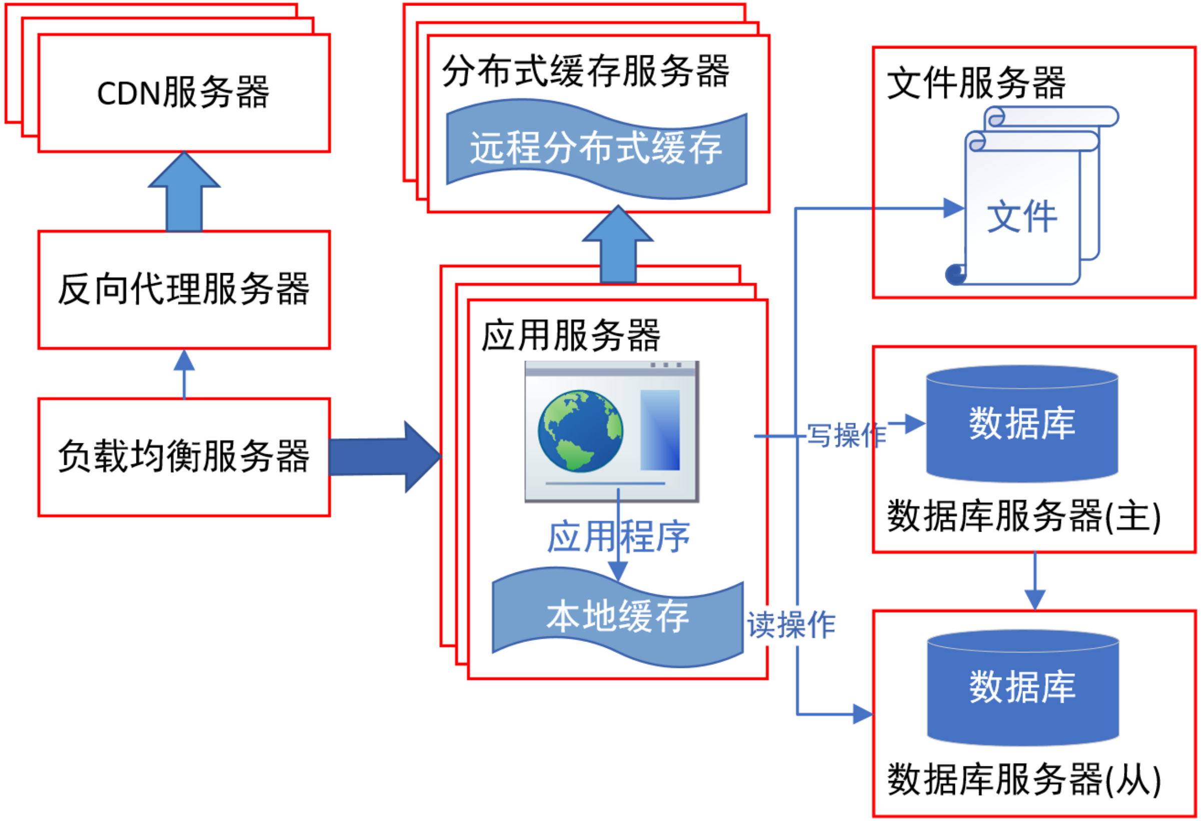
Architecture of Java Web Application

REliable, INtelligent & Scalable Systems

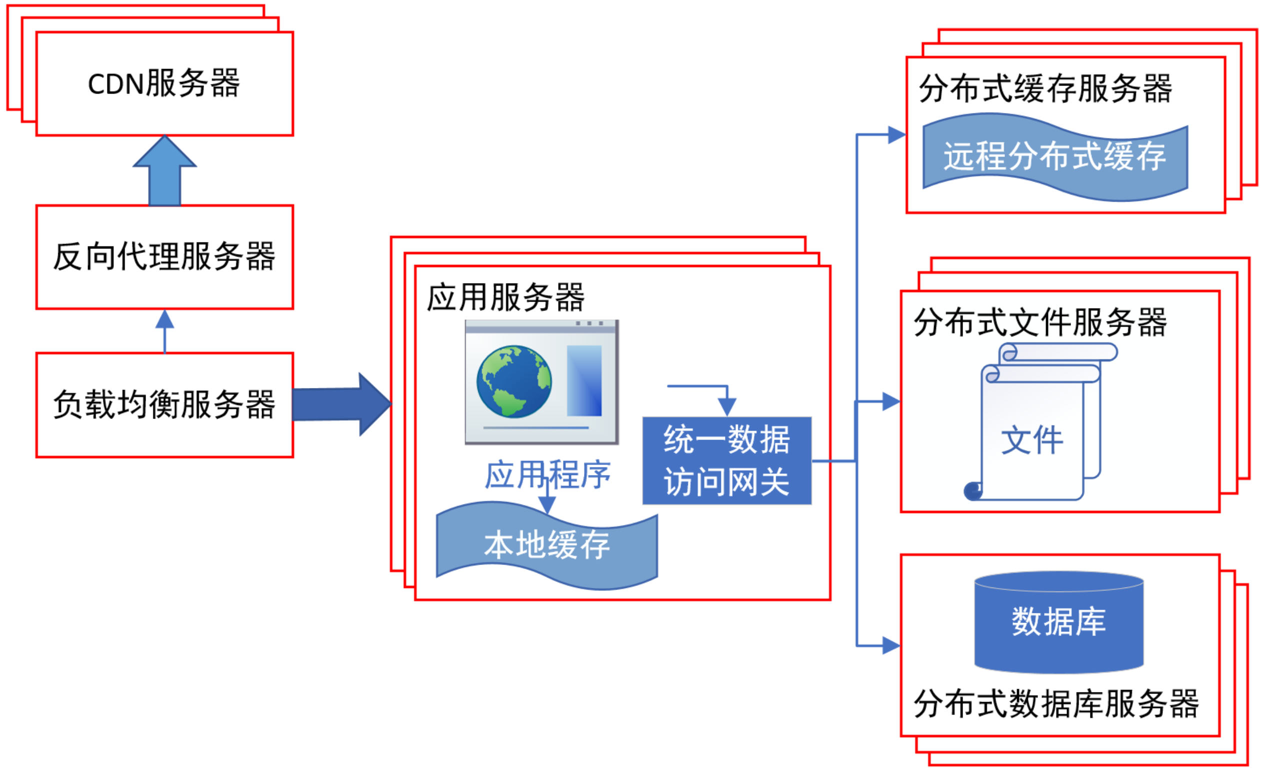


Architecture of Java Web Application

REliable, INtelligent & Scalable Systems

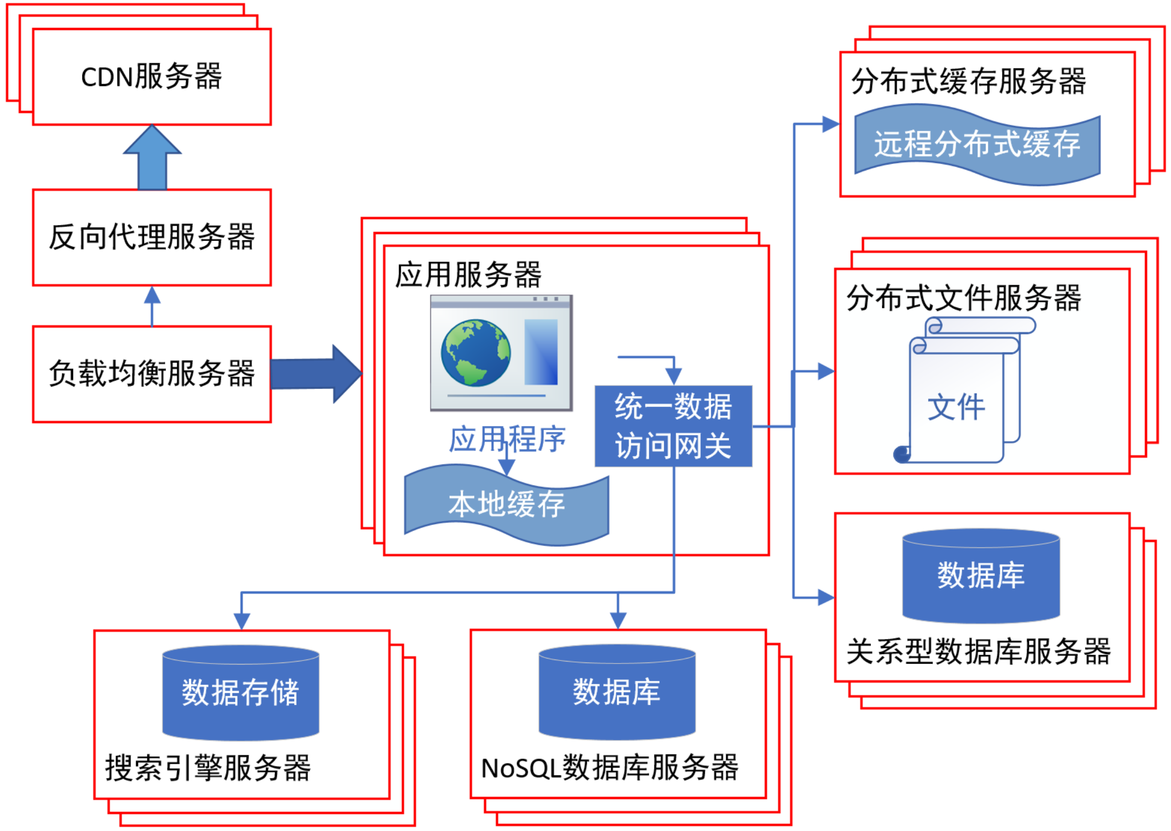


Architecture of Java Web Application



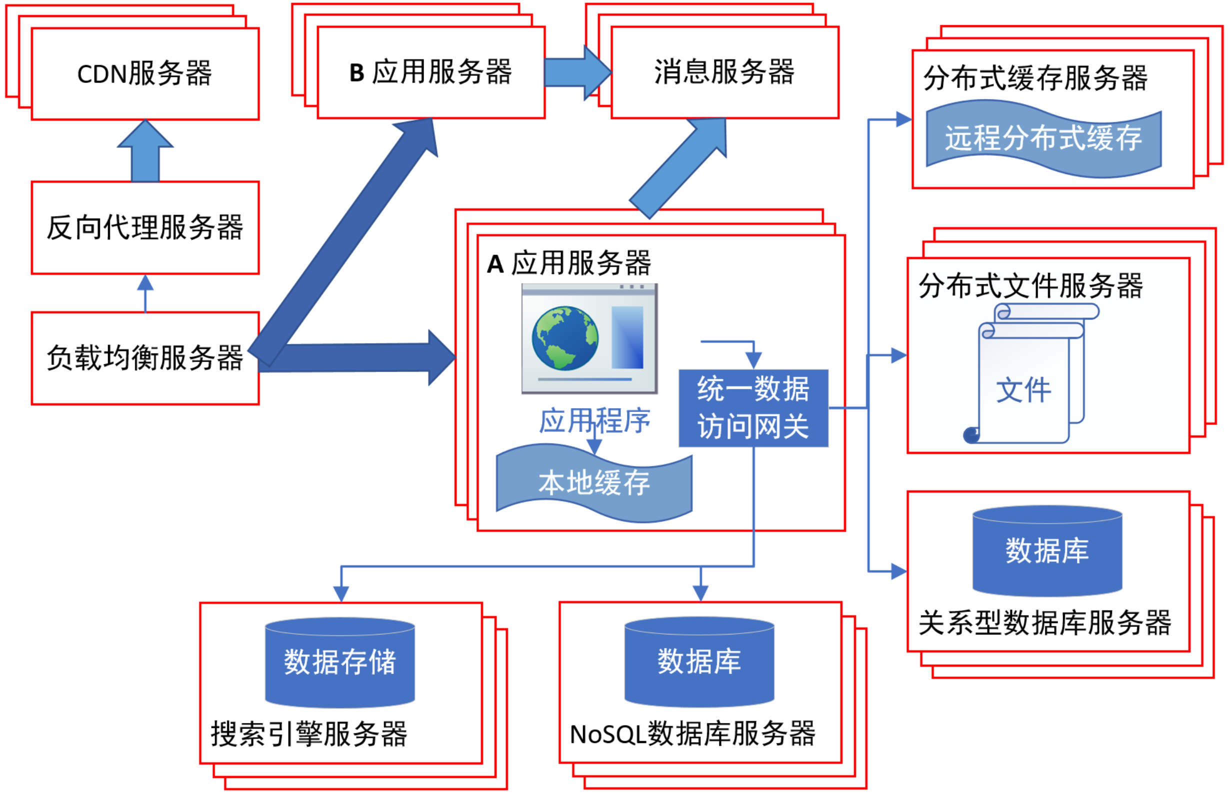
Architecture of Java Web Application

REliable, INtelligent & Scalable Systems



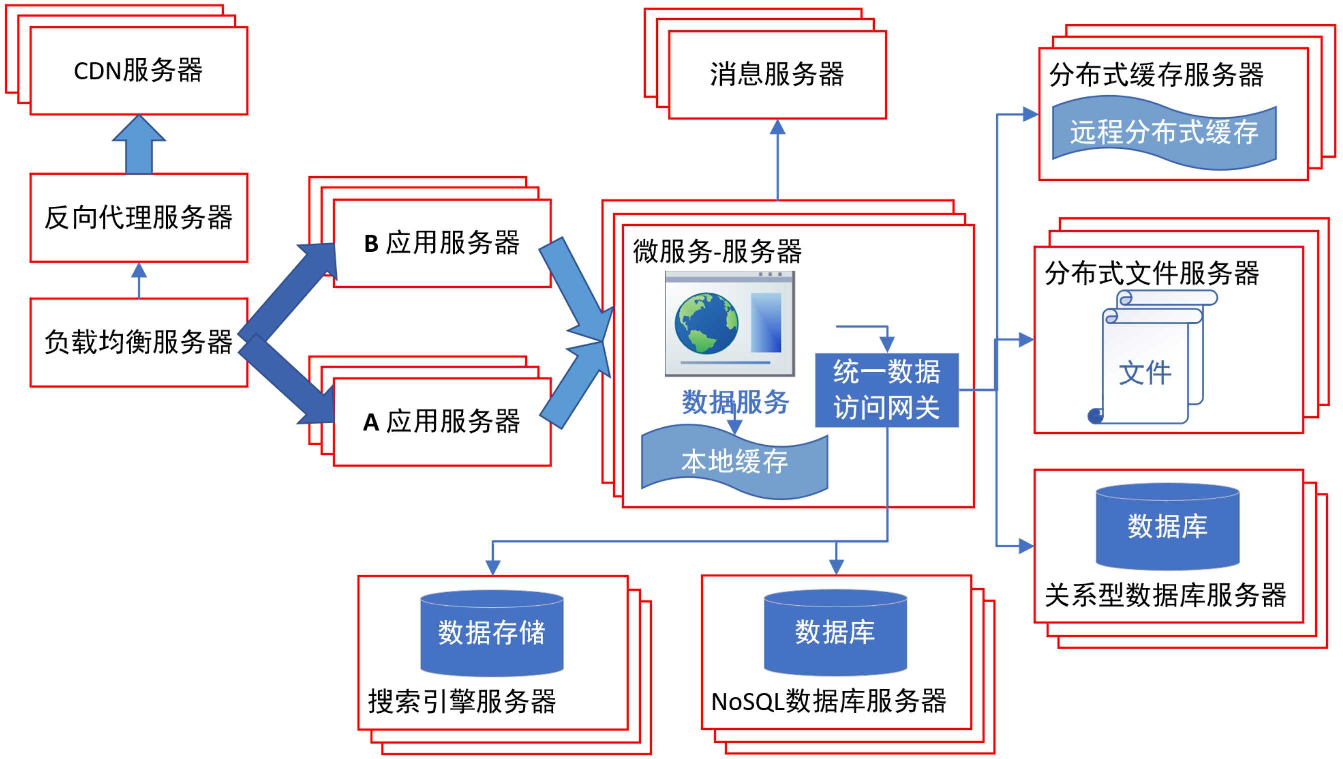
Architecture of Java Web Application

REliable, INtelligent & Scalable Systems



Architecture of Java Web Application

REliable, INtelligent & Scalable Systems



Architecture of Enterprise Applications 1
Stateful and Stateless Services

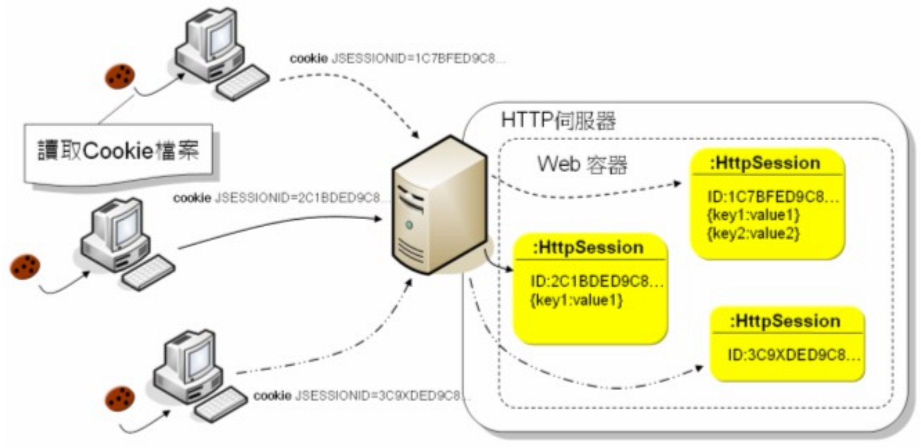
Haopeng Chen

RE*liable,* ***IN****telligent and* ***S****calable Systems Group (****REINS****)*
Shanghai Jiao Tong University
Shanghai, China
http://reins.se.sjtu.edu.cn/~chenhp
e-mail: chen-hp@sjtu.edu.cn

Recall: HTTP session state

REliable, INtelligent & Scalable Systems

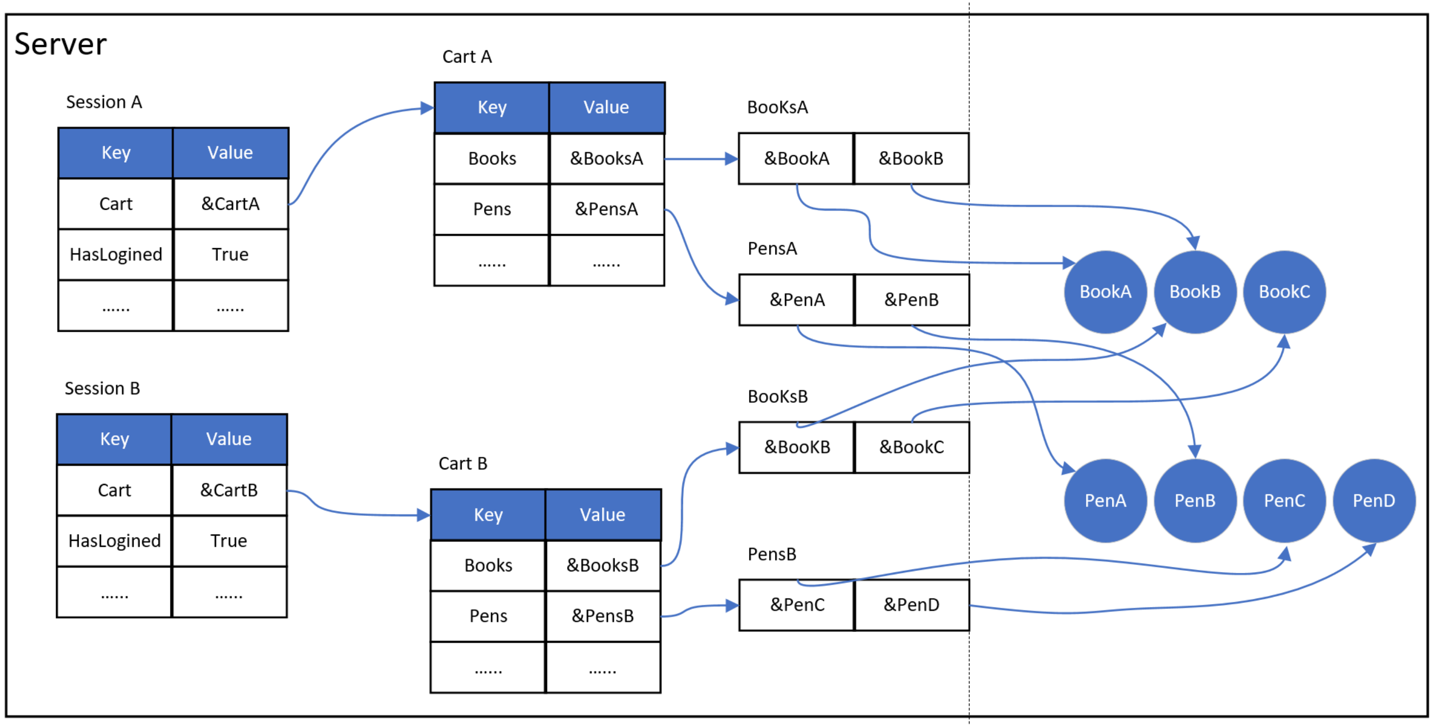
- HTTP is a stateless protocol
  - A stateless protocol does not require the HTTP server to retain information or status about each user for the duration of multiple requests.



  - *from: https://www.openhome.cc/Gossip/ServletJSP/BehindHttpSession.html*

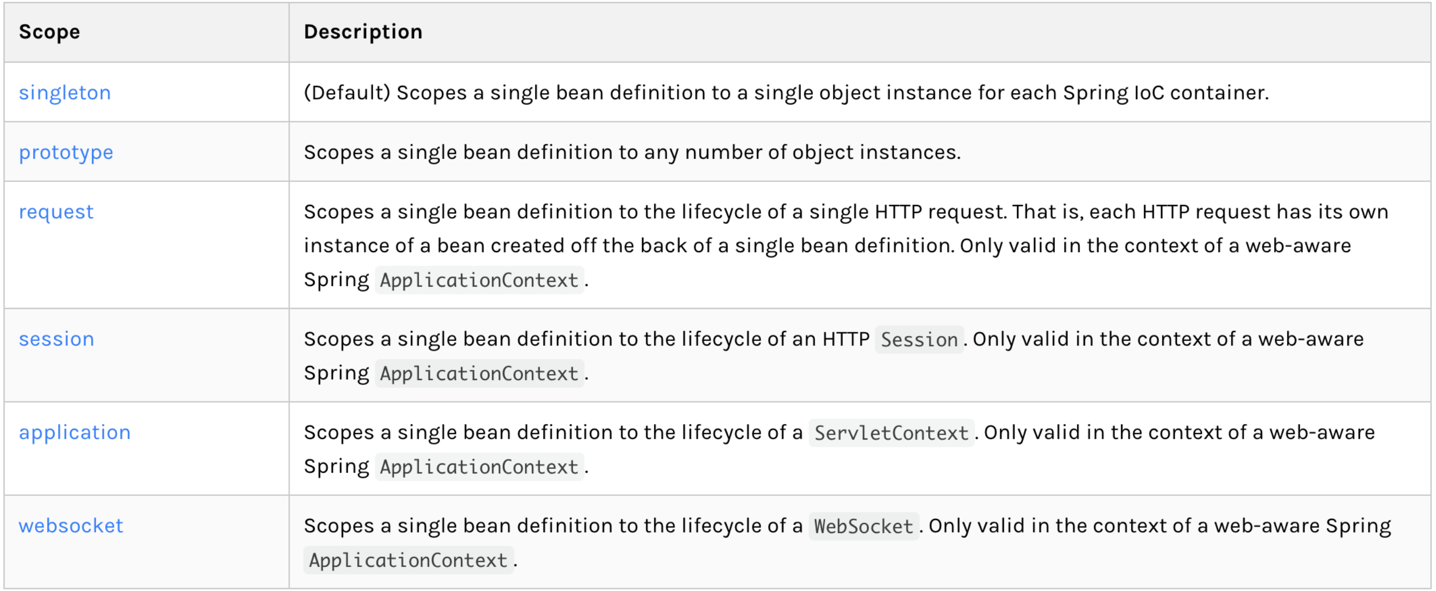
Recall: HTTP session state

- HTTP is a stateless protocol



Spring Bean Scopes

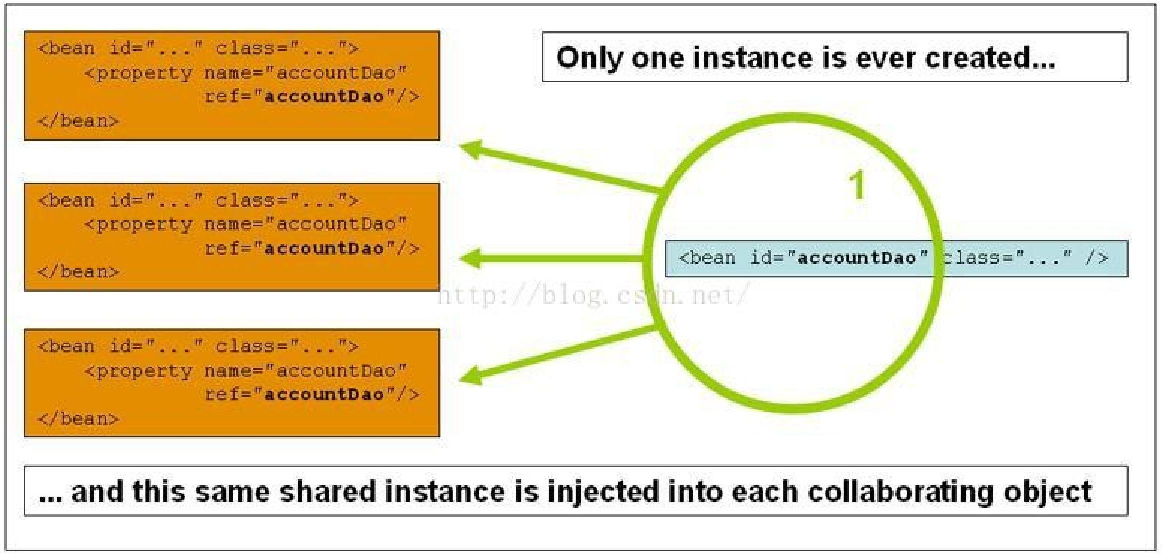
- What’s the meaning of scope?



Scope	Description
singleton	(Default) Scopes a single bean definition to a single object instance for each Spring IoC container.
prototype	Scopes a single bean definition to any number of object instances.
request	Scopes a single bean definition to the lifecycle of a single HTTP request. That is, each HTTP request has its own instance of a bean created off the back of a single bean definition. Only valid in the context of a web-aware Spring `ApplicationContext`.
session	Scopes a single bean definition to the lifecycle of an HTTP `Session`. Only valid in the context of a web-aware Spring `ApplicationContext`.
application	Scopes a single bean definition to the lifecycle of a `ServletContext`. Only valid in the context of a web-aware Spring `ApplicationContext`.
websocket	Scopes a single bean definition to the lifecycle of a `WebSocket`. Only valid in the context of a web-aware Spring `ApplicationContext`.

Scope

- Singleton : Stateless



```
<bean id="..." class="...">
    <property name="accountDao"
              ref="accountDao"/>
</bean>
```

```
<bean id="..." class="...">
    <property name="accountDao"
              ref="accountDao"/>
</bean>
```

```
<bean id="..." class="...">
    <property name="accountDao"
              ref="accountDao"/>
</bean>
```

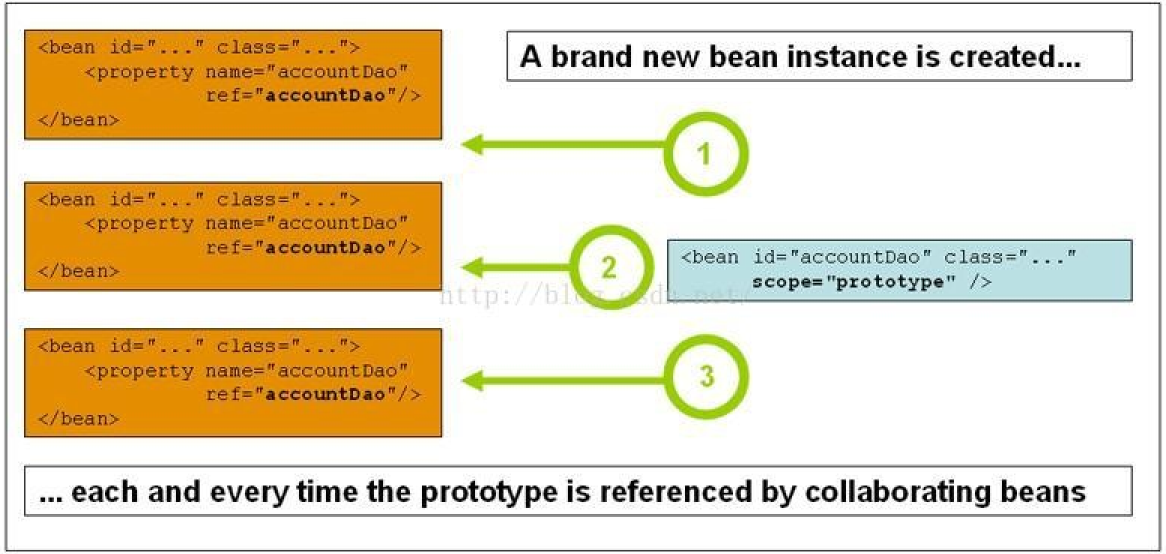
Only one instance is ever created...

```
<bean id="accountDao" class="..." />
```

... and this same shared instance is injected into each collaborating object

Scope

- Prototype : Stateful

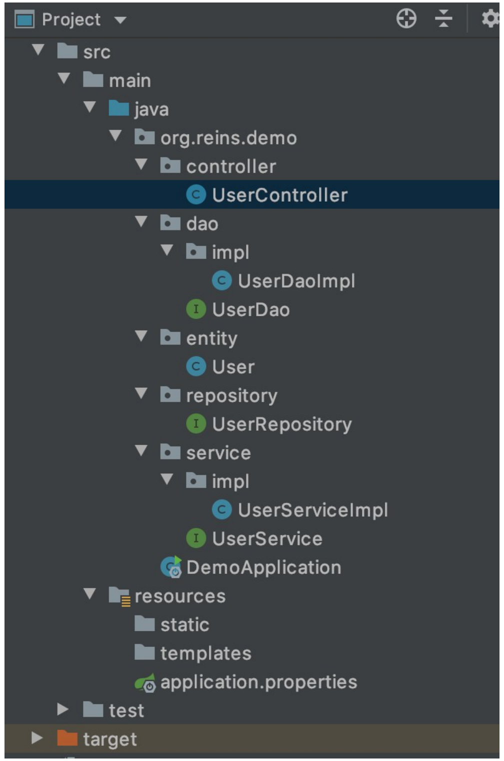

A brand new bean instance is created...
1
2
3
... each and every time the prototype is referenced by collaborating beans

```
<bean id="..." class="...">
    <property name="accountDao"
              ref="accountDao"/>
</bean>
```

```
<bean id="accountDao" class="..."
      scope="prototype" />
```

Default Scope - singleton

REliable, INtelligent & Scalable Systems



```
Project
▼ src
  ▼ main
    ▼ java
      ▼ org.reins.demo
        ▼ controller
            UserController
        ▼ dao
          ▼ impl
              UserDaoImpl
            UserDao
        ▼ entity
            User
        ▼ repository
            UserRepository
        ▼ service
          ▼ impl
              UserServiceImpl
            UserService
          DemoApplication
    ▼ resources
        static
        templates
        application.properties
  ▶ test
▶ target
```

UserController.java

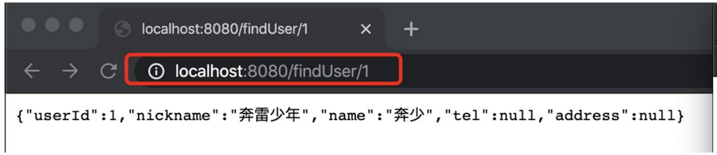
```java
@RestController
public class UserController {

    @Autowired
    private UserService userService;

    @GetMapping(value = "/findUser/{id}")
    public User findOne(@PathVariable("id") String id) {
        System.out.println(userService);
        return userService.findUserById(Integer.valueOf(id));
    };
}
```

UserServiceImpl.java

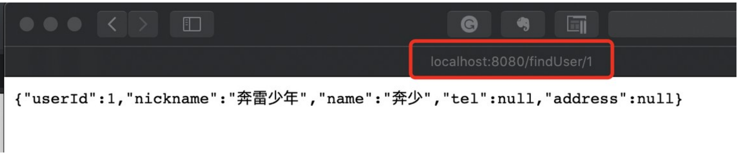
```java
@Service
public class UserServiceImpl implements UserService {
    @Autowired
    private UserDao userDao;

    @Override
    public User findUserById(Integer id) {
        return userDao.findOne(id);
    }
}
```

Default Scope - singleton

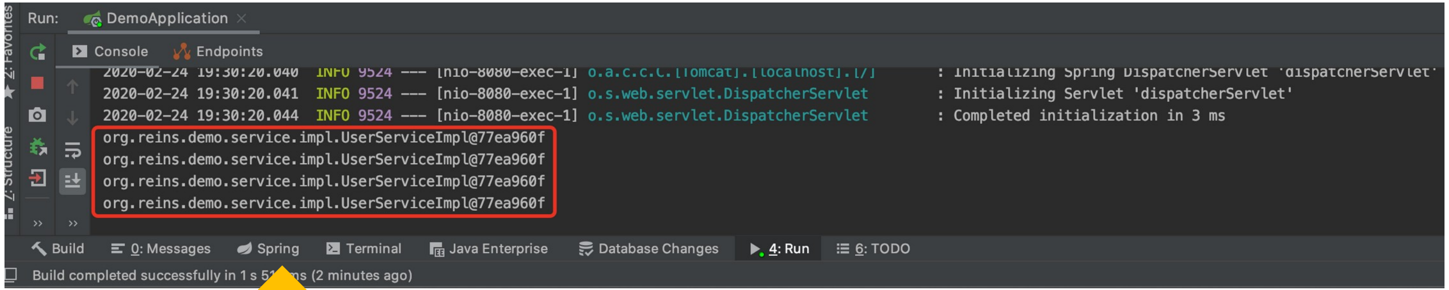
REliable, INtelligent & Scalable Systems



```
{"userId":1,"nickname":"奔雷少年","name":"奔少","tel":null,"address":null}
```



```
{"userId":1,"nickname":"奔雷少年","name":"奔少","tel":null,"address":null}
```



```
2020-02-24 19:30:20.040  INFO 9524 --- [nio-8080-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring DispatcherServlet 'dispatcherServlet'
2020-02-24 19:30:20.041  INFO 9524 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet        : Initializing Servlet 'dispatcherServlet'
2020-02-24 19:30:20.044  INFO 9524 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet        : Completed initialization in 3 ms
org.reins.demo.service.impl.UserServiceImpl@77ea960f
org.reins.demo.service.impl.UserServiceImpl@77ea960f
org.reins.demo.service.impl.UserServiceImpl@77ea960f
org.reins.demo.service.impl.UserServiceImpl@77ea960f
```

Build completed successfully in 1 s 51 ms (2 minutes ago)

Single Service
instance

Scope - prototype

UserController.java

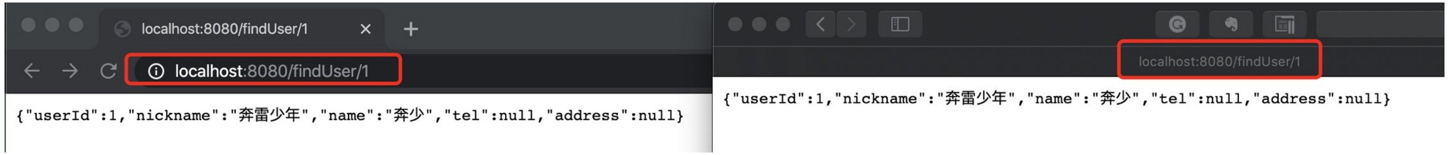
```
@RestController
public class UserController {

    @Autowired
    private UserService userService;

    @GetMapping(value = "/findUser/{id}")
    public User findOne(@PathVariable("id") String id) {
        System.out.println(userService);
        return userService.findUserById(Integer.valueOf(id));
    };
}
```

prototype

UserServiceImpl.java

```
@Service
@Scope("prototype")
public class UserServiceImpl implements UserService {
    @Autowired
    private UserDao userDao;

    @Override
    public User findUserById(Integer id) {
        return userDao.findOne(id);
    }
}
```

Scope - prototype

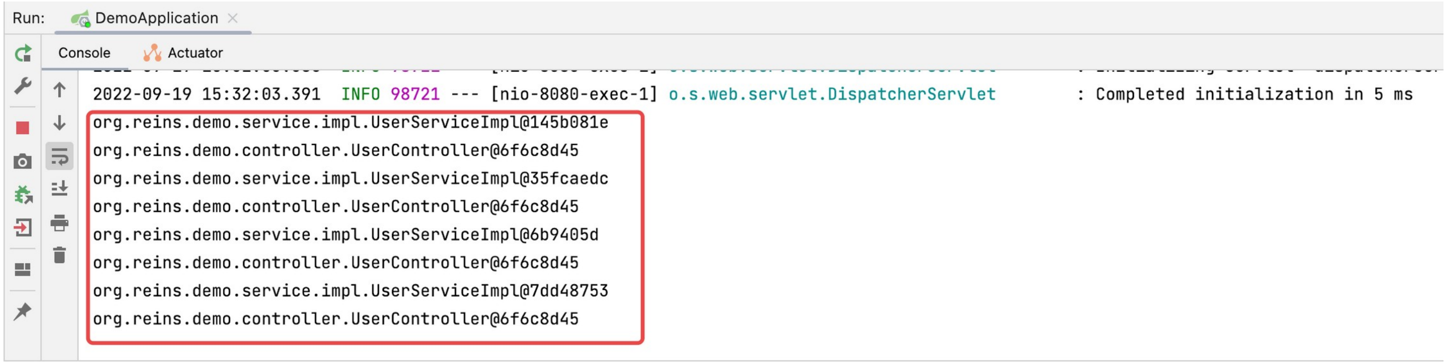


localhost:8080/findUser/1

```
{"userId":1,"nickname":"奔雷少年","name":"奔少","tel":null,"address":null}
```

localhost:8080/findUser/1

```
{"userId":1,"nickname":"奔雷少年","name":"奔少","tel":null,"address":null}
```



```
2022-09-19 15:32:03.391  INFO 98721 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet        : Completed initialization in 5 ms
org.reins.demo.service.impl.UserServiceImpl@145b081e
org.reins.demo.controller.UserController@6f6c8d45
org.reins.demo.service.impl.UserServiceImpl@35fcaedc
org.reins.demo.controller.UserController@6f6c8d45
org.reins.demo.service.impl.UserServiceImpl@6b9405d
org.reins.demo.controller.UserController@6f6c8d45
org.reins.demo.service.impl.UserServiceImpl@7dd48753
org.reins.demo.controller.UserController@6f6c8d45
```

Now, different

Scope - prototype

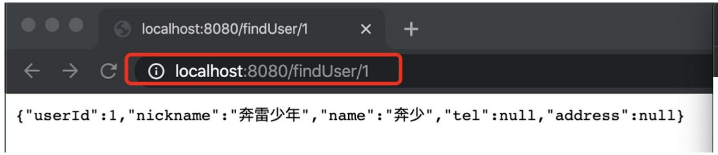
prototype

UserController.java
```
@RestController
@Scope("prototype")
public class UserController {

    @Autowired
    private UserService userService;

    @GetMapping(value = "/findUser/{id}")
    public User findOne(@PathVariable("id") String id) {
        System.out.println(userService);
        return userService.findUserById(Integer.valueOf(id));
    };
}
```

prototype

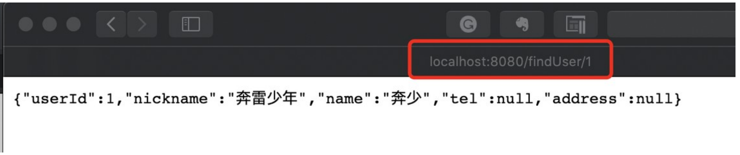
UserServiceImpl.java
```
@Service
@Scope("prototype")
public class UserServiceImpl implements UserService {
    @Autowired
    private UserDao userDao;

    @Override
    public User findUserById(Integer id) {
        return userDao.findOne(id);
    }
}
```

Scope - prototype



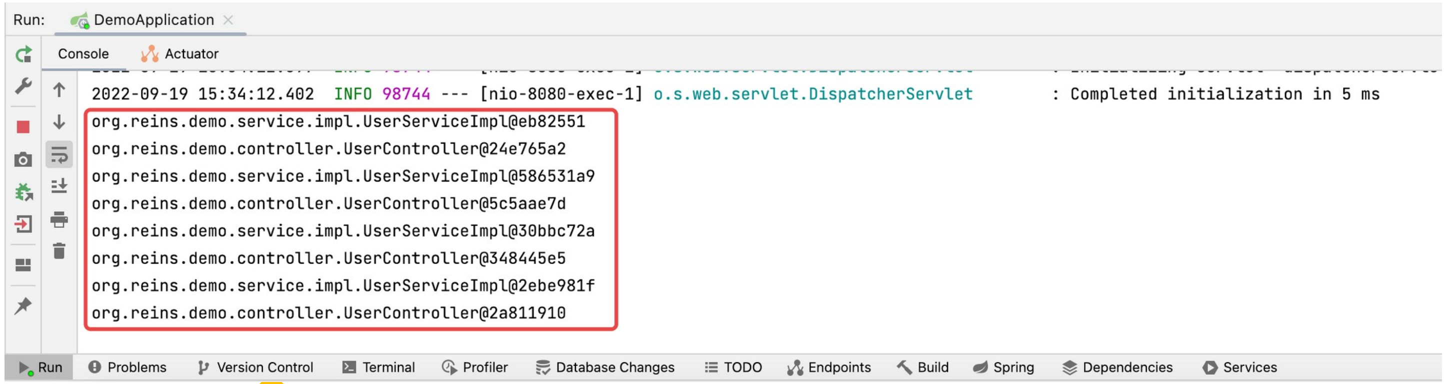
localhost:8080/findUser/1

```
{"userId":1,"nickname":"奔雷少年","name":"奔少","tel":null,"address":null}
```



localhost:8080/findUser/1

```
{"userId":1,"nickname":"奔雷少年","name":"奔少","tel":null,"address":null}
```



```
2022-09-19 15:34:12.402  INFO 98744 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet        : Completed initialization in 5 ms
org.reins.demo.service.impl.UserServiceImpl@eb82551
org.reins.demo.controller.UserController@24e765a2
org.reins.demo.service.impl.UserServiceImpl@586531a9
org.reins.demo.controller.UserController@5c5aae7d
org.reins.demo.service.impl.UserServiceImpl@30bbc72a
org.reins.demo.controller.UserController@348445e5
org.reins.demo.service.impl.UserServiceImpl@2ebe981f
org.reins.demo.controller.UserController@2a811910
```

Now different. But not good!

Scope - prototype

REliable, INtelligent & Scalable Systems

WebApplicationContext

UserServic
e

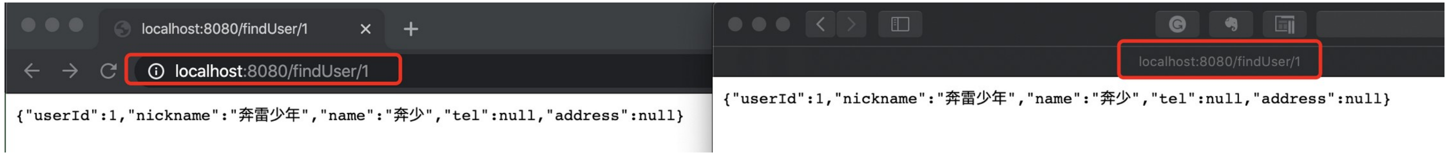
prototype

UserController.java
```
@RestController
public class UserController {
    @Autowired
    WebApplicationContext applicationContext;

    @GetMapping(value = "/findUser/{id}")
    public User findOne(@PathVariable("id") String id) {
        UserService userService = applicationContext.getBean(UserService.class);
        System.out.println(userService);
        return userService.findUserById(Integer.valueOf(id));
    };
}
```

UserServiceImpl.java
```
@Service
@Scope("prototype")
public class UserServiceImpl implements UserService {
    @Autowired
    private UserDao userDao;

    @Override
    public User findUserById(Integer id) {
        return userDao.findOne(id);
    }
}
```

Scope - prototype

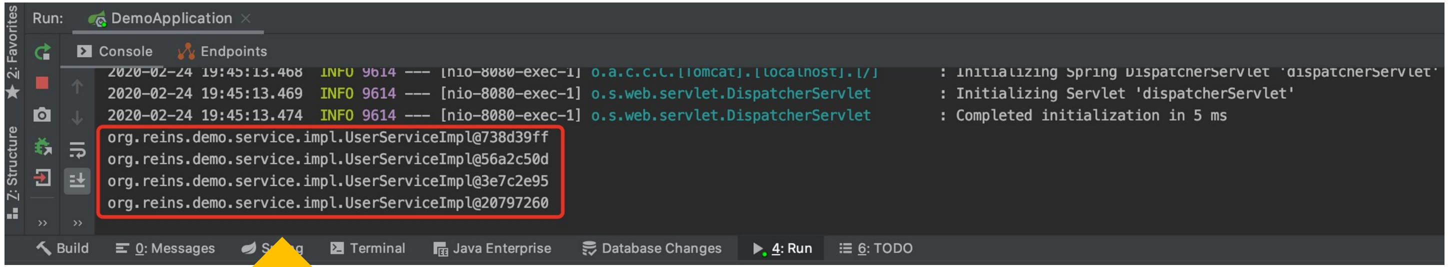
REliable, INtelligent & Scalable Systems



localhost:8080/findUser/1
localhost:8080/findUser/1
localhost:8080/findUser/1

```
{"userId":1,"nickname":"奔雷少年","name":"奔少","tel":null,"address":null}
```

```
{"userId":1,"nickname":"奔雷少年","name":"奔少","tel":null,"address":null}
```



```
2020-02-24 19:45:13.468  INFO 9614 --- [nio-8080-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring DispatcherServlet 'dispatcherServlet'
2020-02-24 19:45:13.469  INFO 9614 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet        : Initializing Servlet 'dispatcherServlet'
2020-02-24 19:45:13.474  INFO 9614 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet        : Completed initialization in 5 ms
org.reins.demo.service.impl.UserServiceImpl@738d39ff
org.reins.demo.service.impl.UserServiceImpl@56a2c50d
org.reins.demo.service.impl.UserServiceImpl@3e7c2e95
org.reins.demo.service.impl.UserServiceImpl@20797260
```

Now different. Why is this solution better?

Scope - session

REliable, INtelligent & Scalable Systems

WebApplicationContext

UserServic
e

session

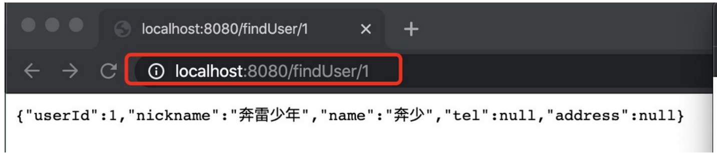
UserController.java

```
@RestController
public class UserController {
    @Autowired
    WebApplicationContext applicationContext;

    @GetMapping(value = "/findUser/{id}")
    public User findOne(@PathVariable("id") String id) {
        UserService userService = applicationContext.getBean(UserService.class);
        System.out.println(userService);
        return userService.findUserById(Integer.valueOf(id));
    };
}
```

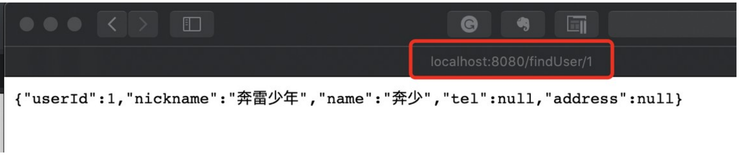
UserServiceImpl.java

```
@Service
@Scope(”session")
public class UserServiceImpl implements UserService {
    @Autowired
    private UserDao userDao;

    @Override
    public User findUserById(Integer id) {
        return userDao.findOne(id);
    }
}
```

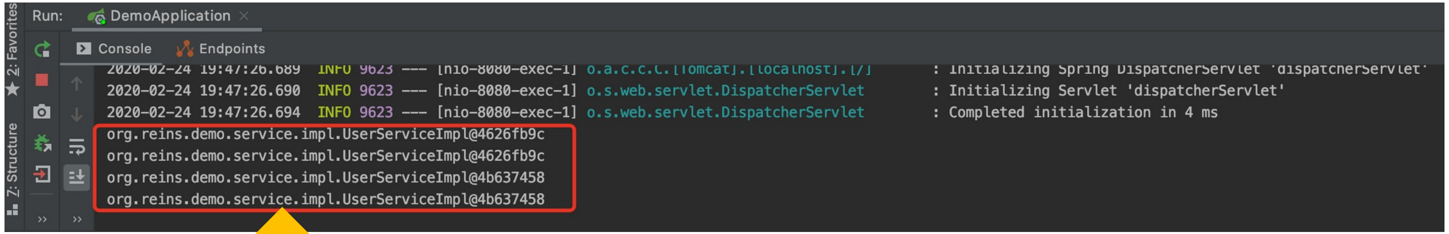
Scope - session



```
{"userId":1,"nickname":"奔雷少年","name":"奔少","tel":null,"address":null}
```



```
{"userId":1,"nickname":"奔雷少年","name":"奔少","tel":null,"address":null}
```



```
2020-02-24 19:47:26.689  INFO 9623 --- [nio-8080-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring DispatcherServlet 'dispatcherServlet'
2020-02-24 19:47:26.690  INFO 9623 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet        : Initializing Servlet 'dispatcherServlet'
2020-02-24 19:47:26.694  INFO 9623 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet        : Completed initialization in 4 ms
org.reins.demo.service.impl.UserServiceImpl@4626fb9c
org.reins.demo.service.impl.UserServiceImpl@4626fb9c
org.reins.demo.service.impl.UserServiceImpl@4b637458
org.reins.demo.service.impl.UserServiceImpl@4b637458
```

Two Service instances

Scope - session

REliable, INtelligent & Scalable Systems

session

WebApplicationContext

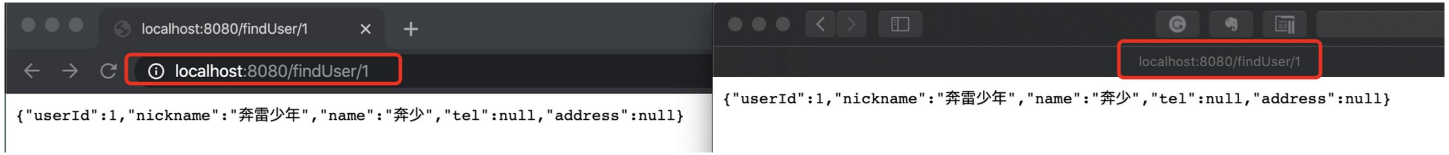
UserServic
e

this

prototype

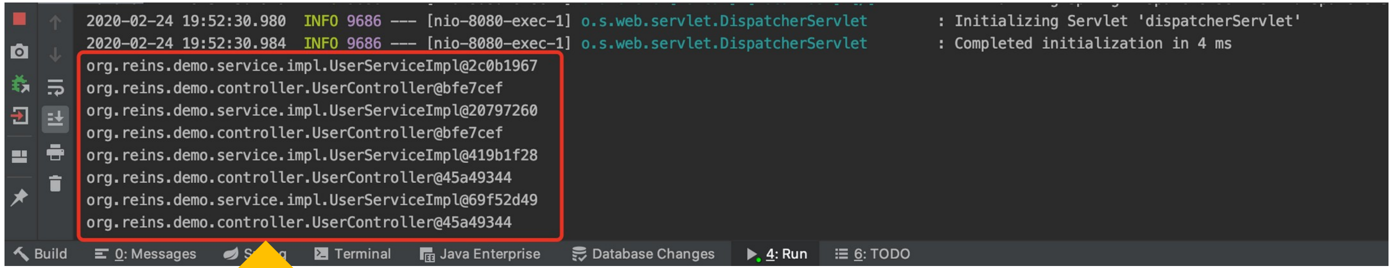
```
UserController.java
@RestController
@Scope(”session")
public class UserController {
    @Autowired
    WebApplicationContext applicationContext;

    @GetMapping(value = "/findUser/{id}")
    public User findOne(@PathVariable("id") String id) {
        UserService userService = applicationContext.getBean(UserService.class);
        System.out.println(userService);
        System.out.println(this);
        return userService.findUserById(Integer.valueOf(id));
    };
}
UserServiceImpl.java
@Service
@Scope(”prototype")
public class UserServiceImpl implements UserService {
    @Autowired
    private UserDao userDao;

    @Override
    public User findUserById(Integer id) {
        return userDao.findOne(id);
    }
}
```

Scope - session


localhost:8080/findUser/1
localhost:8080/findUser/1
localhost:8080/findUser/1

```
{"userId":1,"nickname":"奔雷少年","name":"奔少","tel":null,"address":null}
```

```
{"userId":1,"nickname":"奔雷少年","name":"奔少","tel":null,"address":null}
```



```
2020-02-24 19:52:30.980  INFO 9686 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet        : Initializing Servlet 'dispatcherServlet'
2020-02-24 19:52:30.984  INFO 9686 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet        : Completed initialization in 4 ms
org.reins.demo.service.impl.UserServiceImpl@2c0b1967
org.reins.demo.controller.UserController@bfe7cef
org.reins.demo.service.impl.UserServiceImpl@20797260
org.reins.demo.controller.UserController@bfe7cef
org.reins.demo.service.impl.UserServiceImpl@419b1f28
org.reins.demo.controller.UserController@45a49344
org.reins.demo.service.impl.UserServiceImpl@69f52d49
org.reins.demo.controller.UserController@45a49344
```

Two Controller
instances
Four Service instances

Scope - session

session

WebApplicationContext

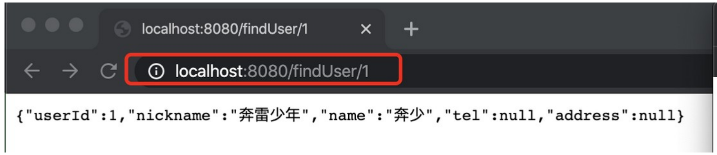
UserService

this

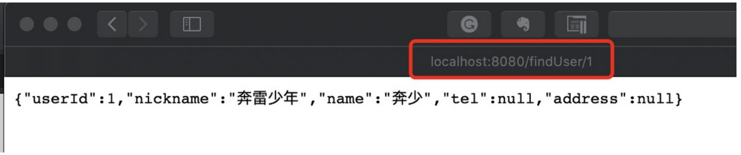
session

```
UserController.java
@RestController
@Scope(”session")
public class UserController {
    @Autowired
    WebApplicationContext applicationContext;

    @GetMapping(value = "/findUser/{id}")
    public User findOne(@PathVariable("id") String id) {
        UserService userService = applicationContext.getBean(UserService.class);
        System.out.println(userService);
        System.out.println(this);
        return userService.findUserById(Integer.valueOf(id));
    };
}
UserServiceImpl.java
@Service
@Scope(”session")
public class UserServiceImpl implements UserService {
    @Autowired
    private UserDao userDao;

    @Override
    public User findUserById(Integer id) {
        return userDao.findOne(id);
    }
}
```

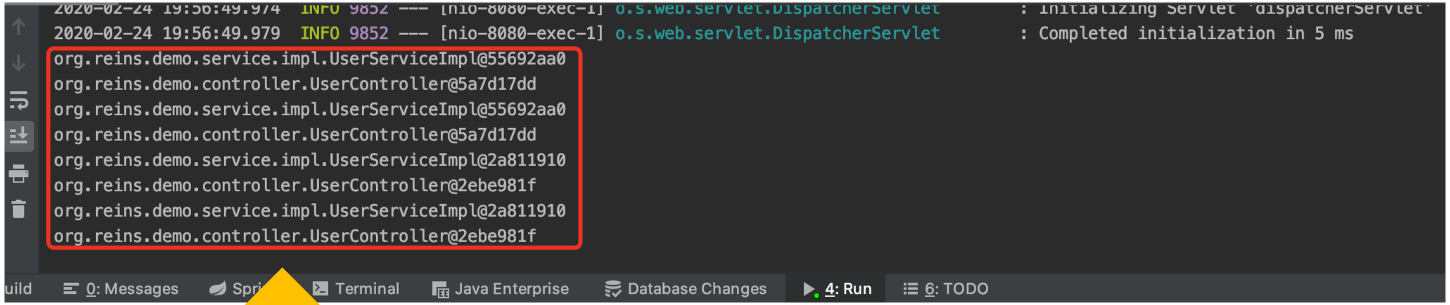
Scope - session



```
{"userId":1,"nickname":"奔雷少年","name":"奔少","tel":null,"address":null}
```



```
{"userId":1,"nickname":"奔雷少年","name":"奔少","tel":null,"address":null}
```



```
2020-02-24 19:56:49.974  INFO 9852 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet        : Initializing Servlet 'dispatcherServlet'
2020-02-24 19:56:49.979  INFO 9852 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet        : Completed initialization in 5 ms
org.reins.demo.service.impl.UserServiceImpl@55692aa0
org.reins.demo.controller.UserController@5a7d17dd
org.reins.demo.service.impl.UserServiceImpl@55692aa0
org.reins.demo.controller.UserController@5a7d17dd
org.reins.demo.service.impl.UserServiceImpl@2a811910
org.reins.demo.controller.UserController@2ebe981f
org.reins.demo.service.impl.UserServiceImpl@2a811910
org.reins.demo.controller.UserController@2ebe981f
```

Two Controller
instances
Two Service instances

Scope - session

prototype

WebApplicationContext

UserServic
e

this

session

UserController.java

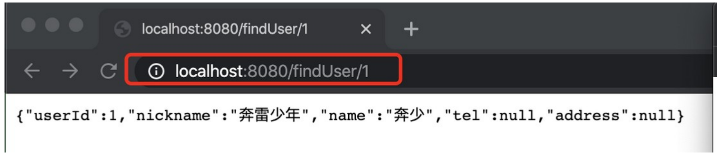
```
@RestController
@Scope(”prototype")
public class UserController {
    @Autowired
    WebApplicationContext applicationContext;

    @GetMapping(value = "/findUser/{id}")
    public User findOne(@PathVariable("id") String id) {
        UserService userService = applicationContext.getBean(UserService.class);
        System.out.println(userService);
        System.out.println(this);
        return userService.findUserById(Integer.valueOf(id));
    };
}
```

UserServiceImpl.java

```
@Service
@Scope(”session")
public class UserServiceImpl implements UserService {
    @Autowired
    private UserDao userDao;

    @Override
    public User findUserById(Integer id) {
        return userDao.findOne(id);
    }
}
```

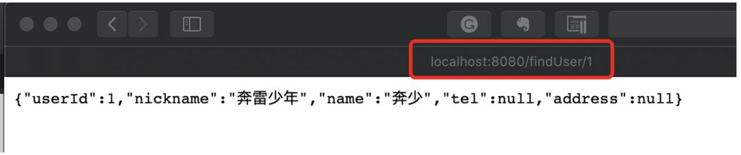
Scope - session

REliable, INtelligent & Scalable Systems



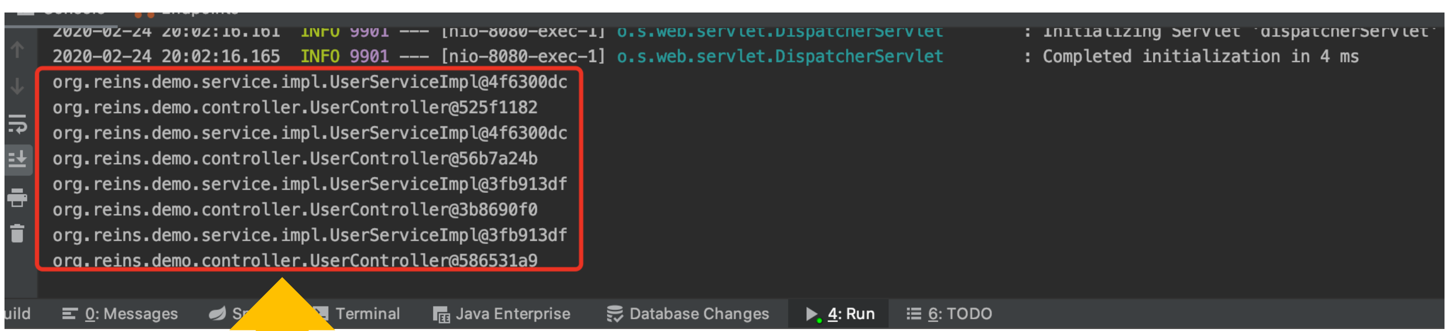
localhost:8080/findUser/1

```
{"userId":1,"nickname":"奔雷少年","name":"奔少","tel":null,"address":null}
```



localhost:8080/findUser/1

```
{"userId":1,"nickname":"奔雷少年","name":"奔少","tel":null,"address":null}
```



```
2020-02-24 20:02:16.161  INFO 9901 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet        : Initializing Servlet 'dispatcherServlet'
2020-02-24 20:02:16.165  INFO 9901 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet        : Completed initialization in 4 ms
org.reins.demo.service.impl.UserServiceImpl@4f6300dc
org.reins.demo.controller.UserController@525f1182
org.reins.demo.service.impl.UserServiceImpl@4f6300dc
org.reins.demo.controller.UserController@56b7a24b
org.reins.demo.service.impl.UserServiceImpl@3fb913df
org.reins.demo.controller.UserController@3b8690f0
org.reins.demo.service.impl.UserServiceImpl@3fb913df
org.reins.demo.controller.UserController@586531a9
```

Four Controller instances
Two Service instances

Summary

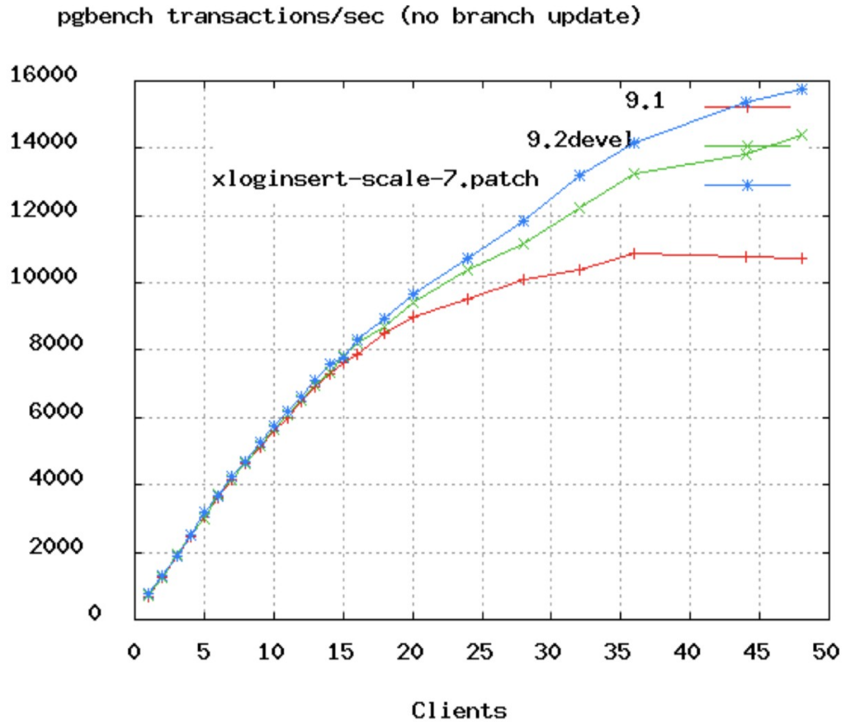
- 2 Browsers(2 Sessions), 4 Requests(2 times / session)

		Controller	
		Prototype	session
Service	prototype	4 service instances 4 controller instances	4 service instances 2 controller instances
	session	2 service instances 4 controller instances	2 service instances 2 controller instances

- How shou

Database Connection Pool Size

- https://github.com/brettwooldridge/HikariCP/wiki/About-Pool-Sizing
- https://www.youtube.com/watch?v=_C77sBcAtSQ



connections = ((core_count * 2) + effective_spindle_count)

Axiom: You want a small pool, saturated with threads waiting for connections.

References

- Web 开发的发展史
  - https://linux.cn/article-3166-1.html
- The IoC Container- Bean Scopes
  - https://docs.spring.io/spring/docs/5.2.3.RELEASE/spring-framework-reference/core.html#beans-factory-scopes
- Quick Guide to Spring Bean Scopes
  - https://www.baeldung.com/spring-bean-scopes
- Spring 注解中 @Scope 的使用解说
  - https://blog.csdn.net/cuichunchi/article/details/79170240

Thank You!