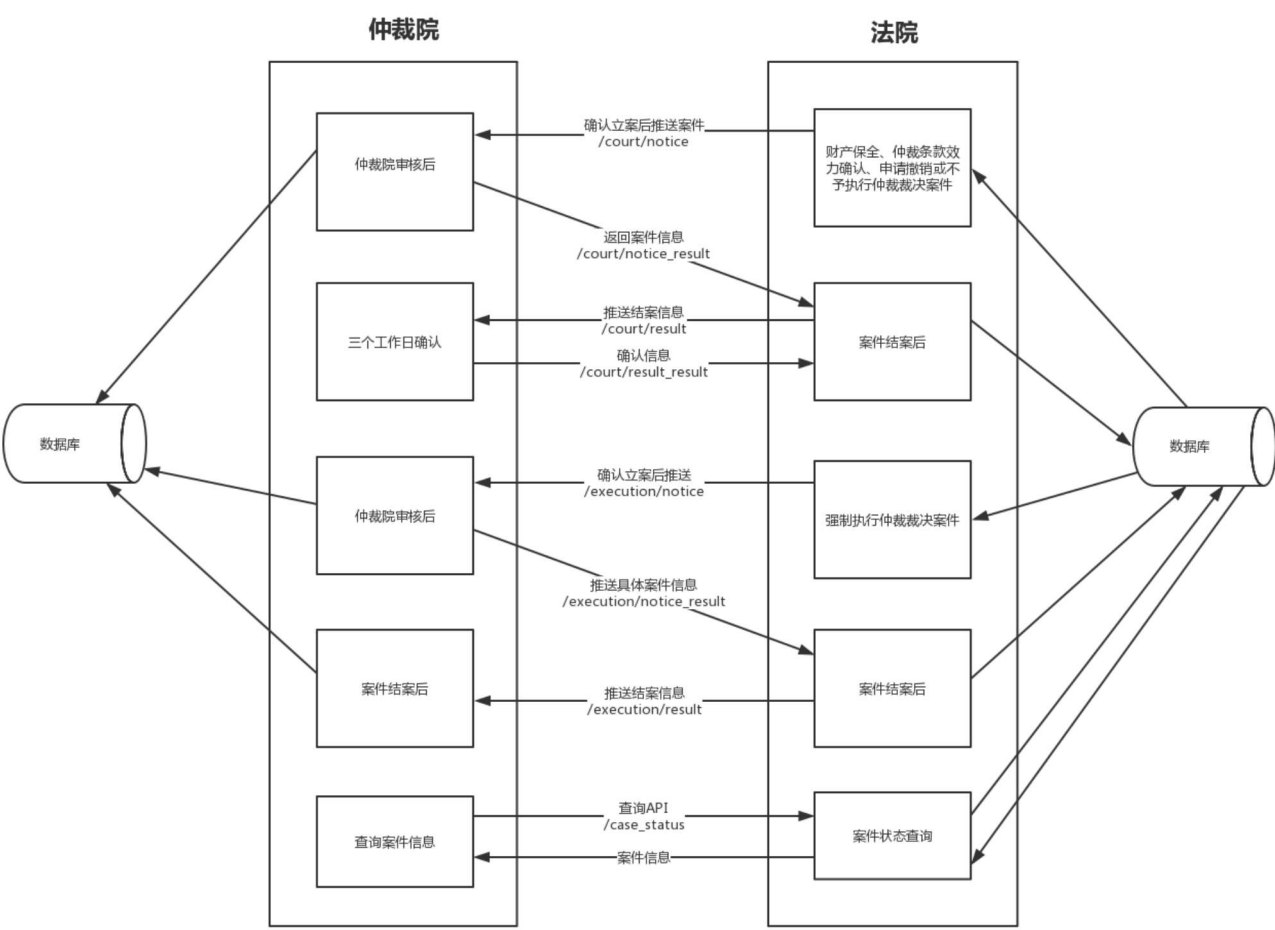


仲裁院和法院项目

仲裁院与法院区别：对应的是仲裁与诉讼

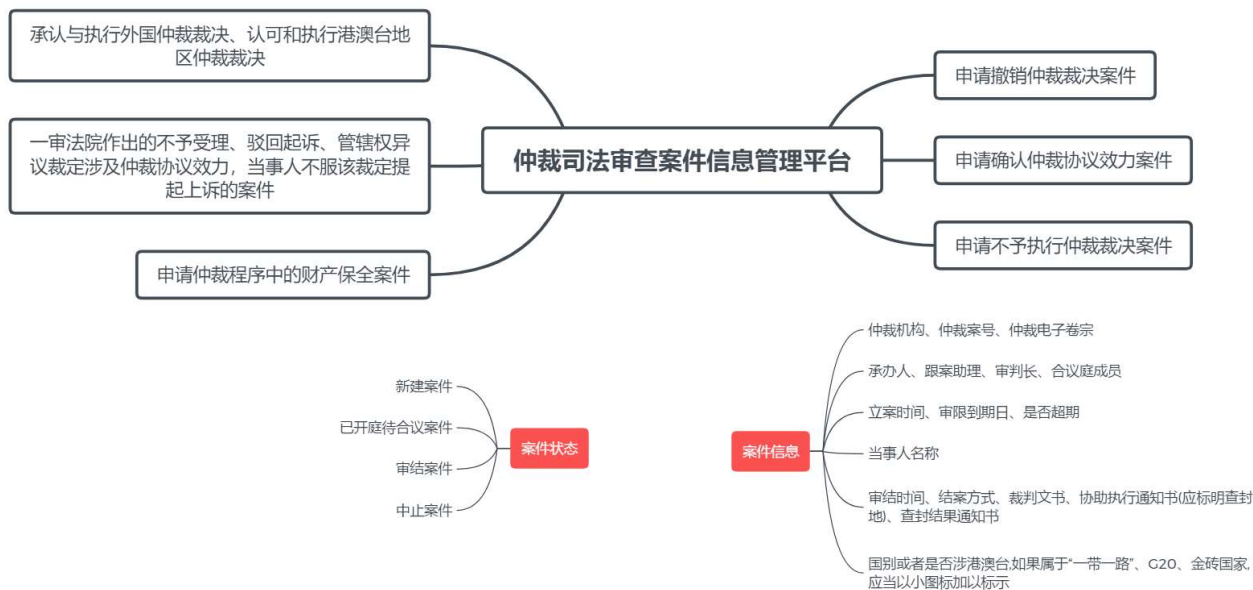
- 1. 仲裁排除法院管辖，即如果选择了仲裁，就不能到法院诉讼了。
- 2. 受案范围不同。有些仲裁是不能受理
- 3. 程序不同，仲裁是一裁终局的，法院诉讼是两审终审。
- 4. 如果债务纠纷 可以到仲裁委员会进行仲裁，有法律效力，如果当事人无力偿还的话，仲裁委员会没有冻结房产和强制执行的权利，可以申请法院强制执行
- 5. 仲裁委员会管辖的原则是双方当事人在合同中约定纠纷由仲裁委员会仲裁，或者纠纷发生后，双方当事人协商一致提交仲裁，否则，只能向法院起诉。双方当事人约定仲裁的，法院就不受理了。

整理出关系



仲裁司法审查案件信息管理平台

整理具体功能



Java反射

类加载器

当Java程序需要使用某个类的时候，如果该类没有被加载到内存中的时候，则系统会通过加载、链接、初始化来对这个类进行初始化

- 加载：将类加载至内存中，并为之创建一个Java对象，任何类在被使用的时候都会创建一个Class对象
- 链接：验证内部正确性，并与其他类协调一致，为类的静态成员分配内存，并设置默认值
- 初始化：类的初始化

类初始化时机

- 创建类的实例
- 访问类的静态方法，为静态变量赋值
- 使用类的静态方法
- 使用反射来强制创建某个类或者接口对应的class类
- 初始化某个类的子类

什么是反射

- 在Java运行状态中，对于任何一个类来说，都能知道这个类的所有的属性和方法；对于任何一个对象，都能够调用任何的方法和属性；这样动态地获取新的以及动态调用对象的方法的功能就叫做反射。
- 简单来说就是解剖一个类，前提是要获得这个类的Class对象

如何使用反射

- 使用Class类，获取出被解剖的这个类的class文件对象
- 使用Class类方法，获取出类中的所有成员
- 将类成员获取出来后，交给对应类、方法执行成员

如何获取Class文件对象

- 使用类的对象获取

```
1 Person p1=new Person();
2 Class<?> cc=p1.getClass();
```

- 使用类的静态属性获取

```
1 Class<?> cc2=Person.class;
```

- 使用Class的静态方法获取（常用来加载数据库驱动）

```
1 Class<?> cc3=Class.forName("com.test.demo.Person"); //全类名
```

Spring

AOP编程

简单理解：执行一些代码前需要执行另外的代码

IOC控制反转

实现：读取标注或者配置文件，看看依赖的是什么，然后拿到类名；使用反射API，基于类名实例化对应的对象实例；将对象实例化，通过构造函数或者setter传递

得到Spring容器对象

- bean工厂，BeanFactory（功能简单）

通过Resource获取BeanFactory

```
1 Resource resource = new ClassPathResource("applicationContext.xml"); // 加载spring配置文件
2 BeanFactory beanFactory = new XmlBeanFactory(resource); // 创建IOC容器对象
```

- 应用上下文，ApplicationContext（功能强大）

通过XML获取ApplicationContext

```
1 ApplicationContext ac = new ClassPathXmlApplicationContext("applicationContext.xml")
```

在Spring中有三种方法配置对象

- 使用XML文件配置

获取IOC容器后，需要在applicationContext.xml中配置信息，使IOC容器根据xml文件来创建对象

```
1 <bean id="user" class="User"/>
```

然后就可以通过IOC容器来创建对象了

```
1 User user = (User)ac.getBean("user"); // 无参构造函数
```

- 使用注解来配置

简化IOC容器配置，注解可以把对象添加到IOC容器中、处理对象依赖关系

步骤：

- 引入context名称空间
- 开启注解扫描器

创建对象以及处理对象的依赖关系注解：

@Component：扫描器

@Configuration：配置类

@Component：指定一个对象加入ioc中

@Repository：在持久层使用

@Service：业务逻辑层使用

@Controller：在控制层使用

@Resource：依赖关系

- 使用JavaConfig来配置

由于Spring的自动装配并不能将第三方的库装配到应用中，需要**显式**装配配置。

编写Java类，使用 @Configuration 标注，并使用此类来创建bean

```
1 @Configuration
2 public class Configuration {
3     // 使用@Bean修饰方法，spring内部会将该对象加入ioc，id默认为方法名
4     @Bean
5     public UserDao userDao() {
6         UserDao userDao = new UserDao();
7         return userDao;
8     }
9 }
```