（；´д｀）ゞ关于各个容器创建及读取方法

1. Bean容器的创建加载：

1.直接读取配置文件：ApplicationContext context=new ClassPathApplicationContext(“str”)

2.读取带有注解的配置信息：AnnotationConfigApplicationContext context=new AnnotationConfigApplicationContext(….class);

3.使用FactoryBean容器：

Resource resource=new ClassPathApplicationContext(“”)

BeanFactory factory=new DefaultListableBeanFactory()

BeanDefinitionReader reader=new XmlBeanDefinitionReader((BeanDefinitionRegister)factory)

reader.loadBeanDefinitions(factory)

1. Mybatis容器创建加载
2. 直接读取配置文件：

SqlSessionFactoryBuilder builder=new SqlSessionFactoryBuilder();  
InputStream inputStream= Resources.getResourceAsStream("mybatisConfig.xml");  
SqlSessionFactory factory=builder.build(inputStream);  
SqlSession sqlSession=factory.openSession(true);  
AccountDao accountDao=sqlSession.getMapper(AccountDao.class);

1. 注解于Mybatis

可以将mybatis的构造sql容器的代码封装成方法，并交由spring管理

@Bean  
 public SqlSessionFactory sqlSessionFactory() throws IOException {  
 InputStream inputStream= Resources.*getResourceAsStream*("mybatis-config.xml");  
 SqlSessionFactory sqlSessionFactory=new SqlSessionFactoryBuilder().build(inputStream);  
 return sqlSessionFactory;  
 }  
 @Bean  
 public UserMapper userMapper(SqlSessionFactory sqlSessionFactory){  
 //相当于定义了一个<mapper/>,mybatis文件中没有mapper标签时才需要这玩意儿  
// sqlSessionFactory.getConfiguration().addMapper(UserMapper.class);  
 SqlSession sqlSession=sqlSessionFactory.openSession();  
 return sqlSession.getMapper(UserMapper.class);  
 }