



O **Profile do Spring** é importante quando queremos que a aplicação se **“comporte”** de maneiras diferentes **em ambientes diferentes**. Dessa maneira podemos criar **Profiles** que serão **alterados por parâmetro** na inicialização da aplicação.

Por exemplo, ambientes de **Desenvolvimento** e de **Produção**. O projeto em **ambiente de produção deve mandar emails reais**, enquanto em desenvolvimento isso não seria tão interessante, então eu poderia ajustar **esse ambiente/Profile para ter um envio simulado**.

Então eu teria um **Mock do notificador de email**:

```
@TipoNotificador(PrioridadeNivel.NORMAL)
@Component
public class NotificadorEmailMock implements Notificador {

    @Override
    public void notificar(Cliente cliente, String mensagem){
        System.out.println("MOCK NOTIFICADOR EMAIL - Notificação deveria ser enviada pa
            "através do email: "+cliente.getEmail() + " - " +mensagem);
    }
}
```

Definindo Profile dos Beans

Posso ajustar então essas classes através da anotação **@Profile(“xxx”)** passando o ambiente que ela deverá ser executada. Dessa maneira posso usar o mock em **“Dev”** e o real em **“Prod”**:

```
@Profile("Dev")
@TipoNotificador(PrioridadeNivel.NORMAL)
@Component
public class NotificadorEmailMock implements Notificador {

    @Override
    public void notificar(Cliente cliente, String mensagem){
        System.out.println("MOCK NOTIFICADOR EMAIL - Notificação deveria ser enviada p
            "através do email: "+cliente.getEmail() + " - " +mensagem);
    }
}

@Profile("Prod")
@TipoNotificador(PrioridadeNivel.NORMAL)
@Component
public class NotificadorEmail implements Notificador {

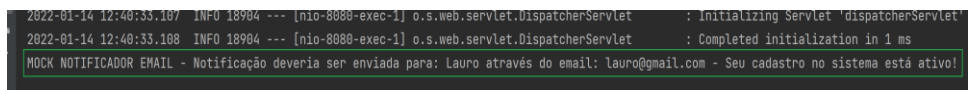
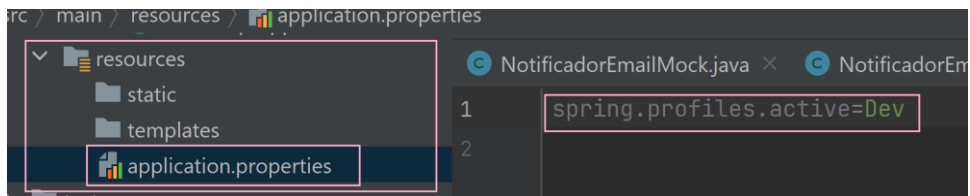
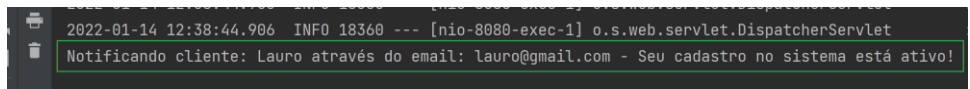
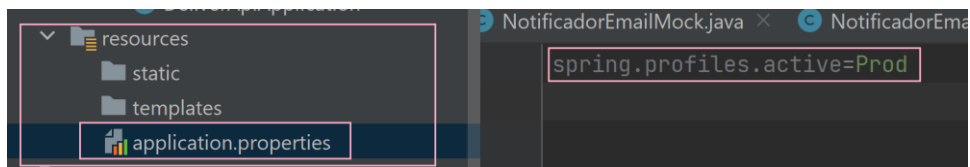
    @Override
    public void notificar(Cliente cliente, String mensagem){
        System.out.println("Notificando cliente: "+cliente.getNome()+" "+
            "através do email: "+cliente.getEmail() + " - " +mensagem);
    }
}
```

Se eu **rodar o projeto com essas anotações eu vou ter um erro** de não encontrar nenhum **Bean** do tipo **Notificador**, isso porque eu **anotei os Profiles mas não iniciei minha aplicação em nenhum deles**, então os **beans** anotados **ficam desconhecidos** para o **container IoC**:

```
treino.deliverapi.servico.AtivarClienteServico required a bean of type 'com.treino.deliverapi.notificador.Notificador' that could not be found.
were found but could not be injected:
```

Definindo Profile da aplicação

Uma das maneiras de definir o Profile da aplicação é usando o arquivo de properties e passando como parâmetro o profile ativo:



Esse profile pode ter **N** parâmetros:

