**Teste no front**

Esse teste foi baseado na PR http://github.amil.com.br/multicanal-prd/credenciado-medico/pull/1946/

Criamos um regra de validação de confirmação de atendimento.

Caso o parâmetro "possuiToken": false e o "caraterAtendimento": "URGENCIA",

Não exibir o card de confirmação de atendimento na página de detalhe do pedido.

Caso o parâmetro "possuiToken": true e o "caraterAtendimento": "URGENCIA",

Exibir o card de confirmação de atendimento na página de detalhe do pedido, apresentando a "data-internacao" e "hora-internacao" do pedido. Não permitir a edição destes valores.

Para a construção de um teste vamos utilizar três arquivos

Os métodos relacionados a **step\_definition.js** são responsáveis pela execução.

**consulta\_pedido\_step\_definition.js**

**confirmar\_atendimento\_step\_definition.js**

**base\_step\_definition.js**

Os dados utilizados na execução estão contidos em **template.js**.

**confirmar-atendimento-template.js**

O roteiro de teste está contido em **.feature** .

**confirmar-atendimento.feature**

Com esse arquivos podemos realizar a implementação do teste

Para a implementar a regra no teste vamos seguir os passos abaixo.

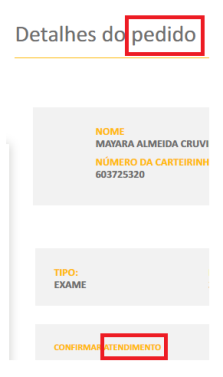
@pedido @atendimento

Cenário: Validar confirmar atendimento com token e carater URGENCIA

1. E consultei o pedido nº 92101863 em situação Autorizado não confirmado
2. E selecionei o resultado
3. E preenchi as informações de Confirmar Atendimento
4. E não preciso preencher as informações de Atendimento
5. Quando clico no botão Confirmar atendimento
6. Então visualizarei a mensagem de Atendimento confirmado
7. E o caso de teste CT005 será contemplado

O Trecho @pedido representa o seguimento do projeto que vamos testar

O Trecho @atendimento representa o card relacionado a nossa regra



No caso 1- (E consultei o pedido nº 92101863 em situação Autorizado não confirmado)utilizamos o código

**this.Given(/^consultei o pedido nº ([^"]\*) em situação ([^"]\*)$/, function (numeroPedido, situacao) {**

**consultaPage.preencher.dados({numeroPedido: numeroPedido});**

**consultaPage.clicar.pesquisar();**

**});**

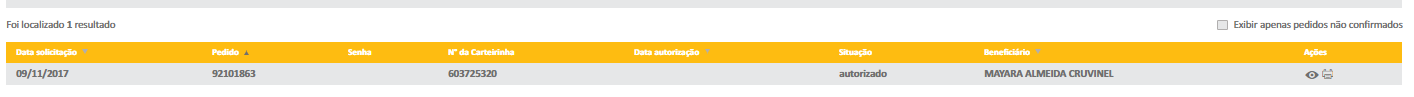


No caso 2- (E selecionei o resultado)

**this.Given(/^(?:selecionei|seleciono) o resultado$/, function () {**

**consultaPage.selecionar.primeiro();**

**});**



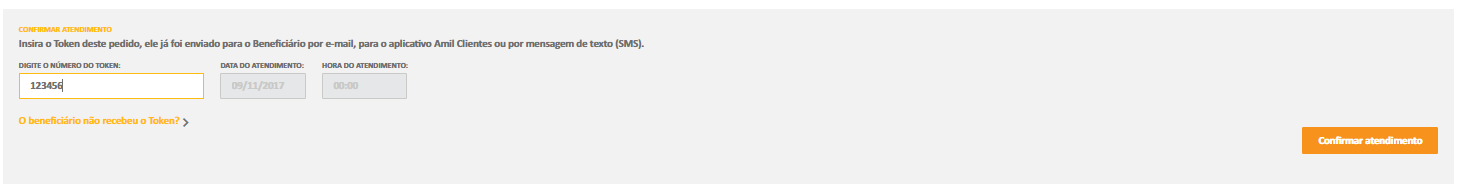
Para os dois casos acima utilizamos a class **consulta\_pedido\_step\_definition.js**

No caso 3-( E preenchi as informações de Confirmar Atendimento)

**this.Given(/^preenchi as informações d[e|o] ([^"]\*)$/, function (tipo) {**

**pedidoPage.preencher.dados(atendimento[tipo]);**

**});**



Para este caso acima utilizamos a class **confirmar\_atendimento\_step\_definition.js**

O campo atendimento está vinculado ao arquivo **confirmar-atendimento-template**

**var atendimento = {**

**'Confirmar Atendimento': {**

**'Digite o número do token:': {**

**value: '12345678',**

**type: 'label'**

**}**

**},**

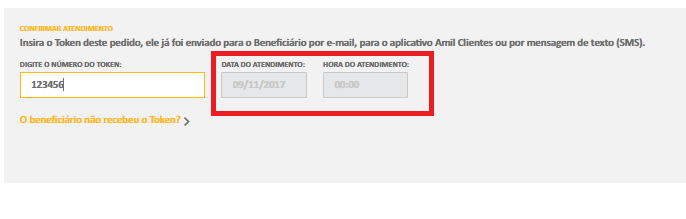
No caso 4-( E não preciso preencher as informações de Atendimento)

**this.Given(/^não preciso preencher as informações de Atendimento$/,function(done){**

**this.expect(pedidoPage.contem.campoDesabilitado('data-atendimento').isPresent()).to.be.eventually.ok.and.notify(done);**

**this.expect(pedidoPage.contem.campoDesabilitado('hora-atendimento').isPresent()).to.be.eventually.ok.and.notify(done);**

**})**

****

Para este caso acima utilizamos a class **confirmar\_atendimento\_step\_definition.js**

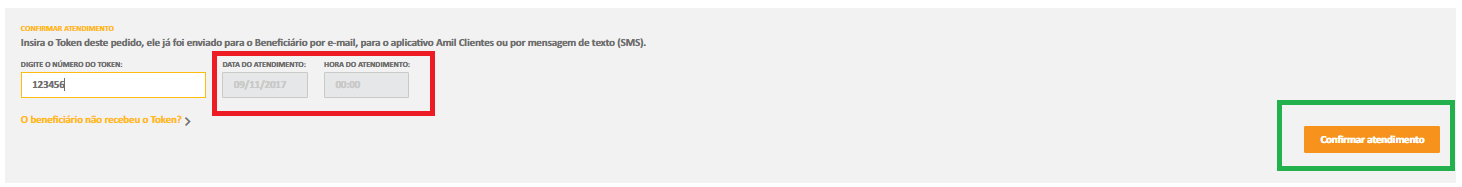
No caso 5- (Quando clico no botão Confirmar atendimento)

**this.When(/^(?:clico|cliquei) no botão ([^"]\*)$/, function (texto, done) {**

**page.clicar.no.botaoLink(texto);**

**done();**

**});**



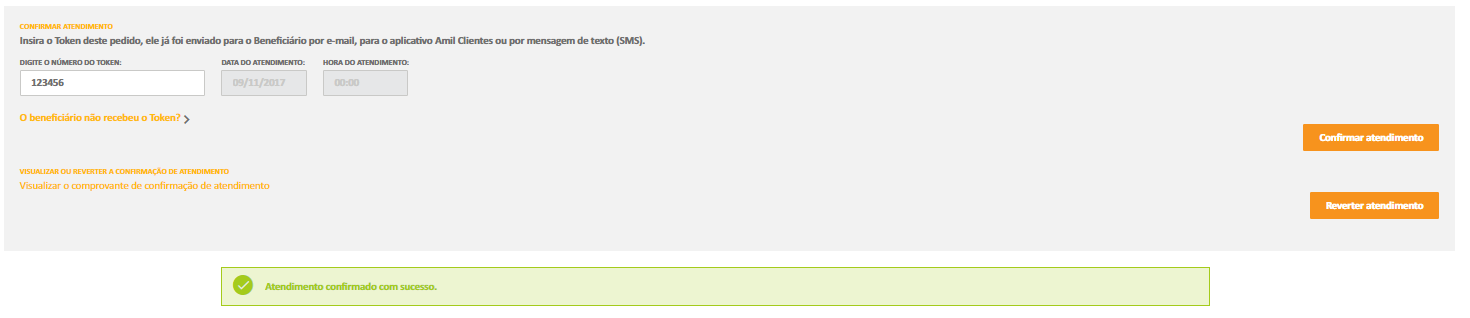
Para este caso acima utilizamos a class **base\_step\_definition.js**

No caso 6- (Então visualizarei a mensagem de Atendimento confirmado)

**this.When(/^visualizarei a mensagem de Atendimento confirmado$/, function (done) {**

**this.expect(pedidoPage.verificar.sucessoConfirmacao()).to.be.eventually.ok.and.notify(done);**

**});**



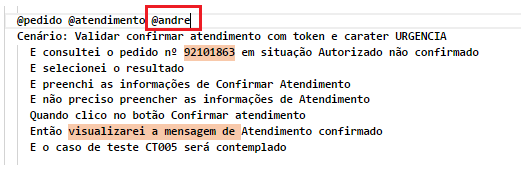
Para este caso acima utilizamos a class **confirmar\_atendimento\_step\_definition.js**

Após implementar todos os passos podemos configurar o teste .

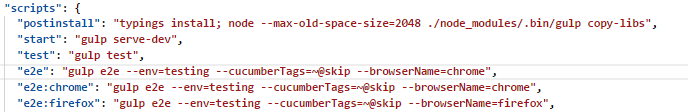
**Configurando o teste no cucumber**

O roteiro de teste está contido em **.feature** .

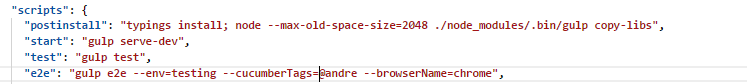
Para indicar a execução de apenas um teste devemos incluir uma tag no cabeçalho do roteiro como indica imagem abaixo.

****

Essa tag será incluída na configuração do cucumber contida no arquivo  da aplicação.



Aqui vamos substituir a tag **~@skip** por **@andre**



Desta forma indicamos ao cucumber que rode apenas os roteiros que conter a tag especificada.

Logo após rode o seguinte comando no terminal ./frontend-test.sh

Será apresentado o seguinte resultado.

