Revisão Certificação Outsystems

Blocos

Evento do bloco não tem de ser configurado se for “não mandatório”

Eventos só existem no contexto dos blocos e estes não têm server actions (não confundir com chamar. O bloco pode chamar mas é uma componente de UI e a das servers é da logica. Se estão em scopes diferentes não faz sentido) (no máximo podem ter data actions)

Um evento pode ser ligado directamente num botão.

As cliente actions tb se chamam screen cliente actions. Podem pregar rasteira em dizer que é triggered no ecrã do pai e isso não é possível.

Entidades Estáticas

Têm atributos e registos. Nas normais os atributos são criados no service studio e os registos a interaccao na app cria os registos.

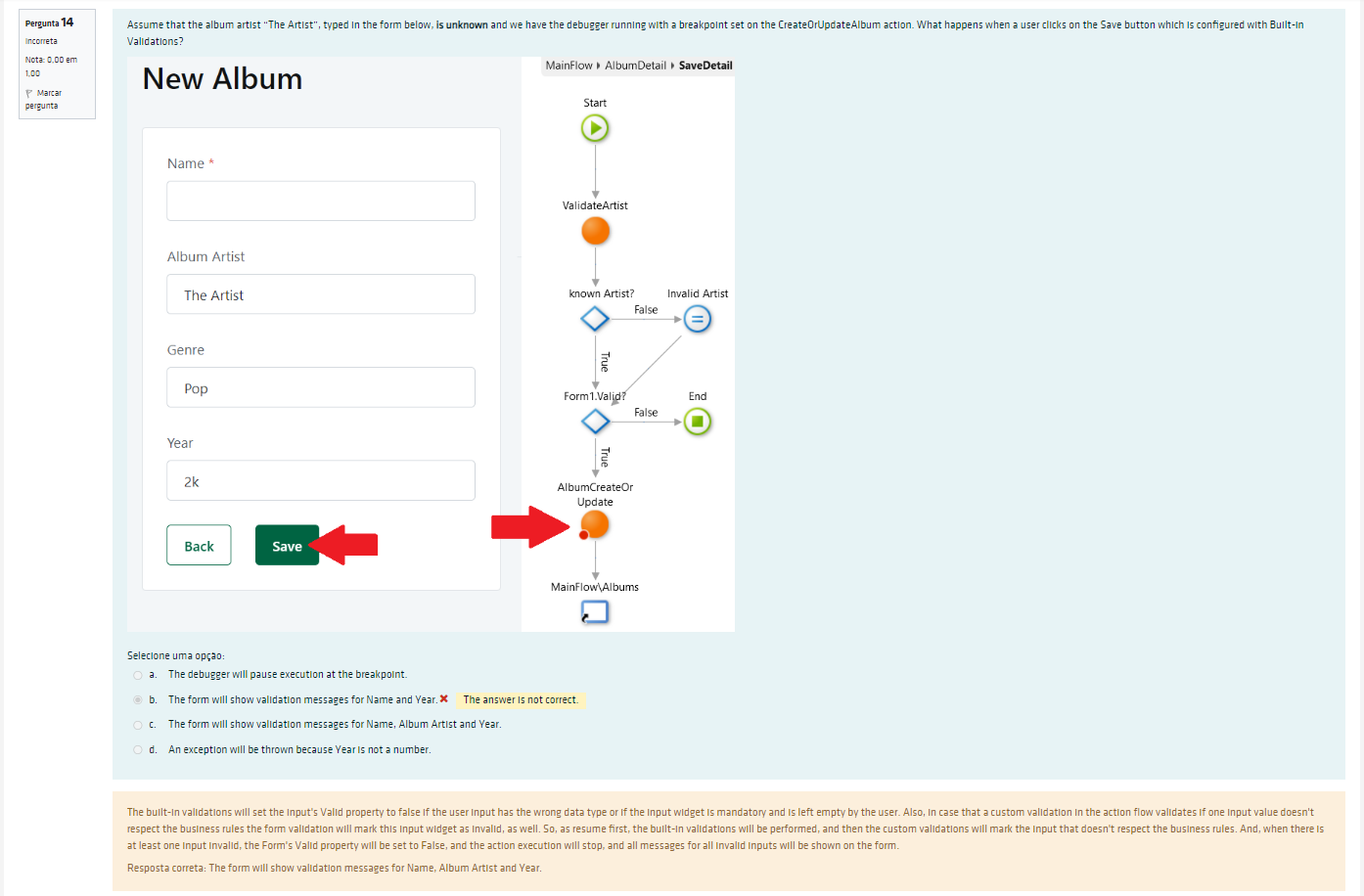
Já as estáticas tudo isso é criado no service studio. Os registos são definidos no service studio e têm um nome.

Entidade Estática não tem acções CRUD. Não podem ser inseridas em runtime usando um create entity action.

O id é autonumber nas estáticas e quando se faz publish se for preciso gerar mais ids isso é feito.

O nome do registo é só um alias para o id. E se passares o rato ele diz “identifier”. Ou seja o nome guarda o registo do id. E o id é autonumber.

Form Validations



Built in validations so vê se os mandatories estao preenchidos e os data types são correctos.

Quando temos built in todos os inputs contribuem para o form valid.

O formulário tem uma coisa que é o valid (um booleano associado ao formulário)

Os inputs tem propriedade valid (booleano) e message (text)

Se um campo de input é mandatorio e não esta preenchido o seu valid passa a falso. Depois o Form valid é composto por

Form.valid = 1.valid AND 2.valid AND 3.valid

Estes 1, 2 e 3 sao os valid dos inputs. E através desta formula vemos que basta um input vir a falso para o form valid passar a falso.

As vezes aparece na certificação uma pergunta destas sem o form valid. Assim vamos directos à base de dados e ela logo diz se pode ser. Neste caso ela não ia curtir e mete o default. No texto ele aceita qualquer coisa.

Não há problema porque não há Foreign Key mandatória. Havendo Foreign Key mandatoria não preenchida, a base de dados ia reclamar.

O mandatorio da base de dados é metadata (informação sobre os outros dados).Tipo o scaffolding saber que aquilo é mandatório e convém trazer para o ecrã.

Em outsystems não há null. Se for mandatorio de texto e for vazio ela vai com “” e está tudo bem.

A diferença entre o mandatorio formulário e a base de dados é que a do form estamos a manipularmos como quisermos. Em termos de base dados a coisa é diferente porque se for mandatoria e for texto, por exemplo, ele coloca “” (coloca sempre o default do data type) e segue.

Normalmente quando um campo é foreign key regra geral metemos dropdown.

O 2k desta pergunta já não seria possível em outsystems porque ao fazer scaffolding ele meteria o input já para numero e nem iria permitir inserir o k.

Debugger

O triangulo faz o trigger da bolinha(exceptions)

O triangulo faz o raise exception e a bolinha é o exception handler (handler tipo nos blocos, o que é que se faz com isto?)

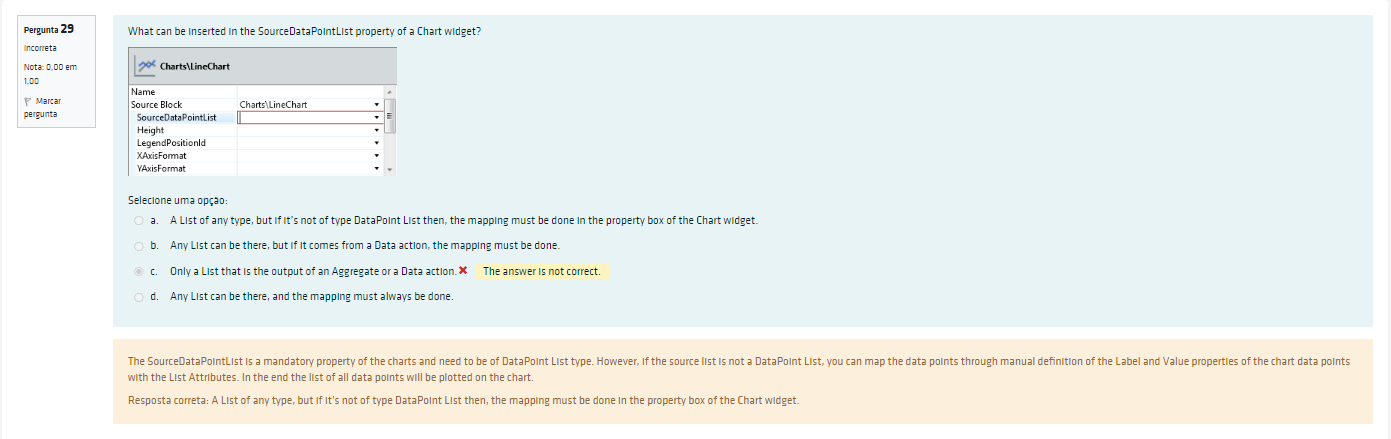
Devemos ter handlers para as excepçoes definidas mas nada nos obriga a ter handlers nas excepçoes. Se não houver handler vai haver https.

Todos os inputs podem ser definidos como mandatorios ou não. (blocos, ecrãs, clientes, servers)

Para handlers do bloco só posso usar acçoes do ecrã pai. Tem a ver com scope. Nunca poderia ir buscar coisas de logica.

Sort dinâmico e paginação vem sempre.

Widgets



Pergunta do line chart.

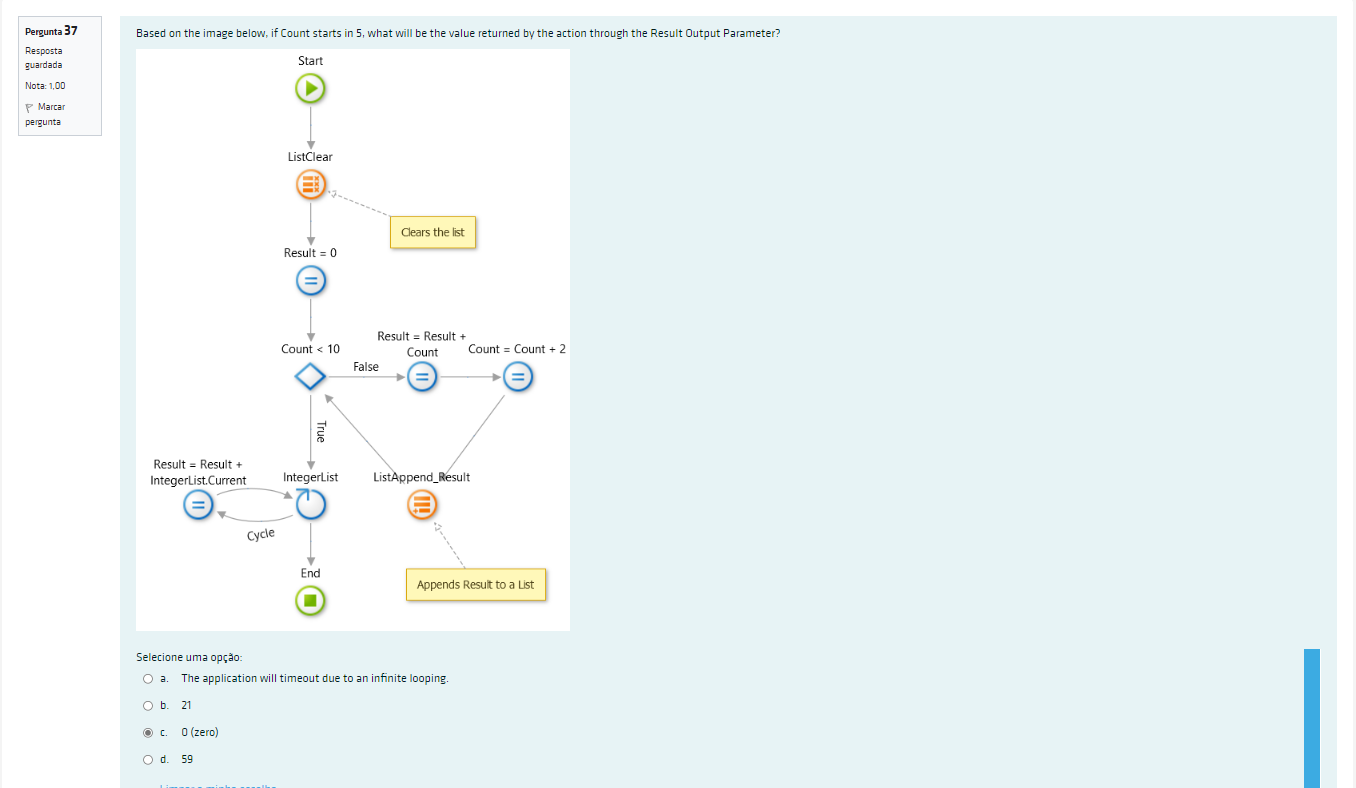
Noçao para cada tipo de widget o que é que ele tem de ter preenchido.

Qualquer chart so funciona se receber uma lista de data points. Se usar uma lista de um agregado faço um mapping.

Mapping tem a ver com o tipo de lista e não de onde vem.

Se for DataPoint list ele não abre o mapeamento mas podemos sempre forçar (mesmo que seja de uma data point list). Se não for DataPointList ele abre automaticamente o mapeamento

Pergunta de Cycle



Aqui na verdade a lista nem entrava no for each porque esta vazia. Nem corria a lista.

Pergunta de servers em funções

Quando não é função pode ter ou não ter outputs.

Se activarmos a função property ela passa a poder ser usada em expressões e passa a ter só um output.

Só pode ter um output mas podemos criar uma estrutura com tudo o que preciso e dizer que o output da função é do tipo da estrutura. É a maneira de ele devolver mais que um valor com um output só.

Falsa seria a que diz que por ser função não pode ser usada num action flow. O facto de ser tornada em função não influencia o seu comportamento como server. Pode continuar a ser usada noutras servers, pode ser publica e usada em action flows.

Do lado do cliente posso arrastar a server action para o action flow. Feito através do web service (do tipo REST). Se a server estiver como função não a podemos usar num assign do ecrã (usá-la como expressão) porque como esta a trabalhar com javascript não faz o webservice (do tipo REST).

Ou seja, do lado do cliente so chamamos acçoes (mesmo que sejam funções). A função directamente pela expressão não consigo usar.

Login

Regarding Login:

Authentication is lost when the application’s request reach the server after session timeout

Users are logged in until they log out

Its not possible to use the login in reactive applications

Its done in the setsessionlogin action of the login screen

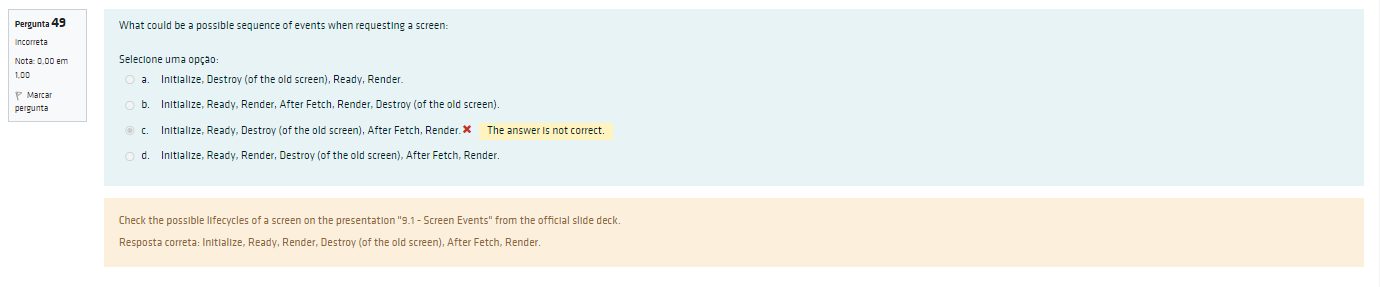
Onde é feito o login? O pedido é feito do lado do cliente mas a autenticação é feita do lado do servidor

|  |  |
| --- | --- |
| Client | Server |
| Feito o pedido (user) | Vai aos users e ve se é match de username e pass |
| Username fica guardado numa local variable do lado do cliente (do login server action) |  |
|  |  |
|  |  |
|  |  |

Todas as chamadas ao servidor do lado do cliente são feitas através de REST web service

Ver notas do caderno

Life cicles



Tenho um input que decide se uma variável que eu tenho vem a true or a false. Isto será definido no Initialize. Podemos ir ao servidor nesta fase mas não é boa prática.

Ready é quando o DOM (documento com tudo o que é preciso para a app correr bem no browser, maestro das diferentes coisas) esta completamente carregado, pronto, done. Se precisarmos de fazer alterações fazemos aqui.

A seguir vem o primeiro render do display que é estático (não depende da base de dados e do servidor)

Nesta pergunta estávamos a falar de uma navegação, então estes 3 primeiros eventos estavam se a referir ao ecrã novo.

Depois do render, como o utilizador já esta a olhar para o ecrã novo, já se pode dar o Destroy do ecrã antigo.

Depois disso se houverem chamadas ao servidor da se o after fetch e SE OS DADOS ALTERAREM ALGUMA COISA NO ECRÃ, o respectivo Render

O primeiro render acontece sempre porque é do conteúdo estático. Se nada mudar no ecrã durante a utilização não ocorrem mais renders.

Estes eventos são todos do lado do cliente (fundo branco)

O ecrã inicia com o “no records to show” porque o agregado começa com os defaults antes de os preenchermos. Ou seja, devemos ter um if com a propriedade empty a dizer que se estiver vazio diz que não há records. Se formos ao servidor e voltarmos com ele igual ele continua vazio, a mensagem continua a mesma e nada renderiza.

O evento on click acontece mesmo quando não o definimos. Tipo clicar no ecrã num sitio em branco. O evento acontece so que como não lhe dissemos para fazer nada, nada acontece.

Assincrono vs Sincrono

Agregada e Data Action são assíncronas e o refresh tb faz com que seja assíncrono. Se enfiar um refresh data action dentro de uma screen cliente action e logo a seguir o seu output pode dar porcaria. Porque como a data action é assíncrona nunca saberei se ao usa-lo num if a seguir ao refresh, se os dados já chegaram.

“Se fizer um data refresh posso fazer ao agregado ou à data action? Aos dois”

No servidor executo tudo de forma síncrona. E a data action vai ao servidor.

O pedido da data action é assíncrono mas tudo o que é feito com esse pedido do lado do servidor é síncrono.

Refresh de agregados não da para fazer no servidor.

Timers não são síncronos e usam coisas do servidor. Excepção!

Exceptions

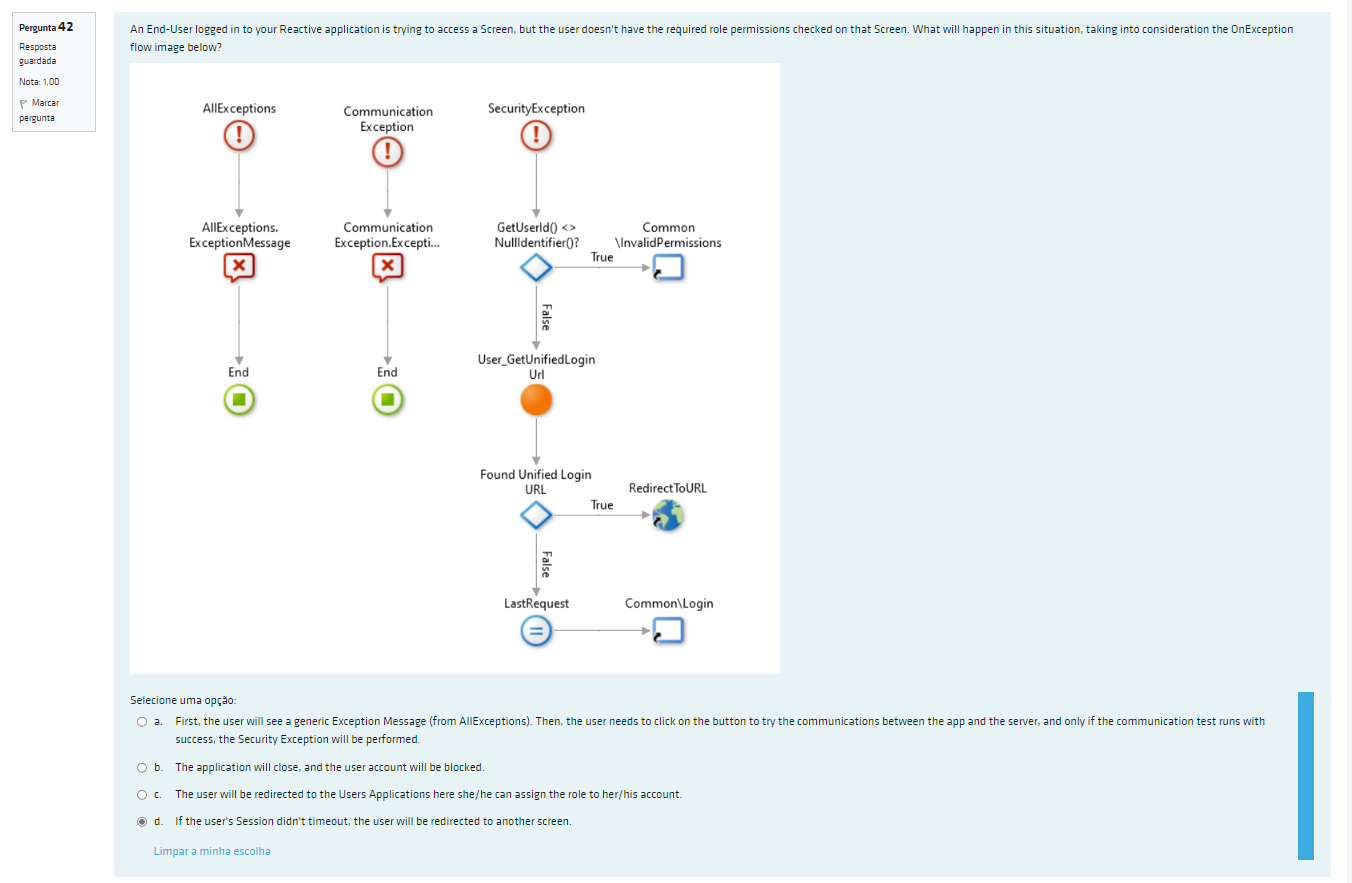
**User exceptions** nunca são automáticas porque somos nos que a criamos e temos de as ir la meter.

As **Database Exception** são sempre automáticas porque é a base de dados que decide se vai levantar erro ou não.

**Security Exceptions** podem ser automáticas ou não. É automático quando um utilizador quer entrar num ecrã sem permissões. Mas nada nos impede de nos metermos um raise dessa exceptions onde quisermos.

**Communication Exception** são sempre automáticas.

A de base de dados e communication são sempre automáticas, as de user nunca são automáticas e as security são mistas.



Nunca se executam dois handlers para a mesma excepção.