

Integrando o cadastro das promoções com o envio

Precisamos enviar uma promoção no formato Json para o browser. Na classe `Promo`, adicione o método `toJson()`, para construir um Json a partir dos dados da `promo`:

```
public class Promo {

    private String titulo;
    private Livro livro = new Livro();

    public String toJson() {
        JsonObjectBuilder promo = Json.createObjectBuilder();
        promo.add("titulo", titulo)
            .add("livroId", livro.getId());

        return promo.build().toString();
    }

    // ...
}
```

Para salvar a lista com os usuários que estão na home da casa do código, crie a classe `UsuariosSession` no pacote `br.com.casadocodigo.loja.websockets`:

```
package br.com.casadocodigo.loja.websockets;

@ApplicationScoped
public class UsuariosSession {

    private List<Session> sessions = new ArrayList<>();
}
```

Adicione o método `add()`, que recebe um `Session`, e o adiciona na lista.

Em seguida adicione o `getUsuarios()`, que retorna a lista de sessões:

```
@ApplicationScoped
public class UsuariosSession {

    private List<Session> sessions = new ArrayList<>();

    public void add(Session session) {
        sessions.add(session);
    }

    public List<Session> getUsuarios() {
        return sessions;
    }
}
```

No `PromosEndpoint`, receba a classe `UsuariosSession` via injeção de dependências:

```
public class PromosEndpoint {  
  
    @Inject  
    private UsuariosSession usuarios;  
  
    // ...  
}
```

Crie o método `send()`, que recebe uma `Promo`. Mova o código que antes ficava no `onMessage()` para esse método. Itere sobre a lista de sessions, disponibilizada pelo método `getUsuarios()`, da classe `UsuariosSession`:

```
public void send(Promo promo) {  
    List<Session> sessions = usuarios.getUsuarios();  
  
    for (Session session : sessions) {  
        if(session.isOpen()) {  
            try {  
                session.getBasicRemote().sendText(promo.toJson());  
            } catch (IOException e) {  
                e.printStackTrace();  
            }  
        }  
    }  
}
```

Faça com que o método `onMessage()` adicione a `Session` recebida na lista do `UsuariosSession`, por meio de uma chamada ao método `add()`.

Por fim, no `AdminPromosBean`, receba o `PromosEndpoint` via injeção de dependências, e faça com que o método `enviar()` chame o método `send()` do `PromosEndpoint`, passando a `promo` que foi cadastrada.

Reinicie o servidor, abra o cadastro de promoções, abra a home da casa do código em uma aba anônima ou em um outro navegador, e tente cadastrar uma nova promoção. Tudo já deve funcionar.