Webアプリケーション開発演習

フィルターって何ぞ？

前回までの学習内容

前回まで前期の復讐をしてきました

- **Webとは？（サーブレットの決まりやセッション）**
- **MVCとは？**

  Vで入力して、Cでチェック、加工を行い

  Mで処理をして、結果をVで表示する。
- **DBとは？**
  - **接続して**
  - **SQLを組み立てて**
  - **SQLを実行して**
  - **結果を返す**

いよいよ、新しいことを学びますっ！

フィルターを理解する！

目次

- フィルターとは？
- フィルターの使用例
- 演習

目次

- フィルターとは？
- フィルターの使用例
- 演習

フィルターとは？

まずは、こちらをご覧ください

フィルターとは？

文字化けしたよね？

フィルターとは？

```
@WebServlet("/page2")
public class Page2Servlet extends HttpServlet {

    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {

        request.setCharacterEncoding("UTF-8");

        String name =  est.getParameter("name");

        request.setAtti        ", name);

        RequestDispatcher                request.getRequestDispatcher("WEB-INF/jsp/page2.jsp");
        dispatcher.forward                );
    }
}
```

```
request.setCharacterEncoding("UTF-8");
```

フィルターとは？

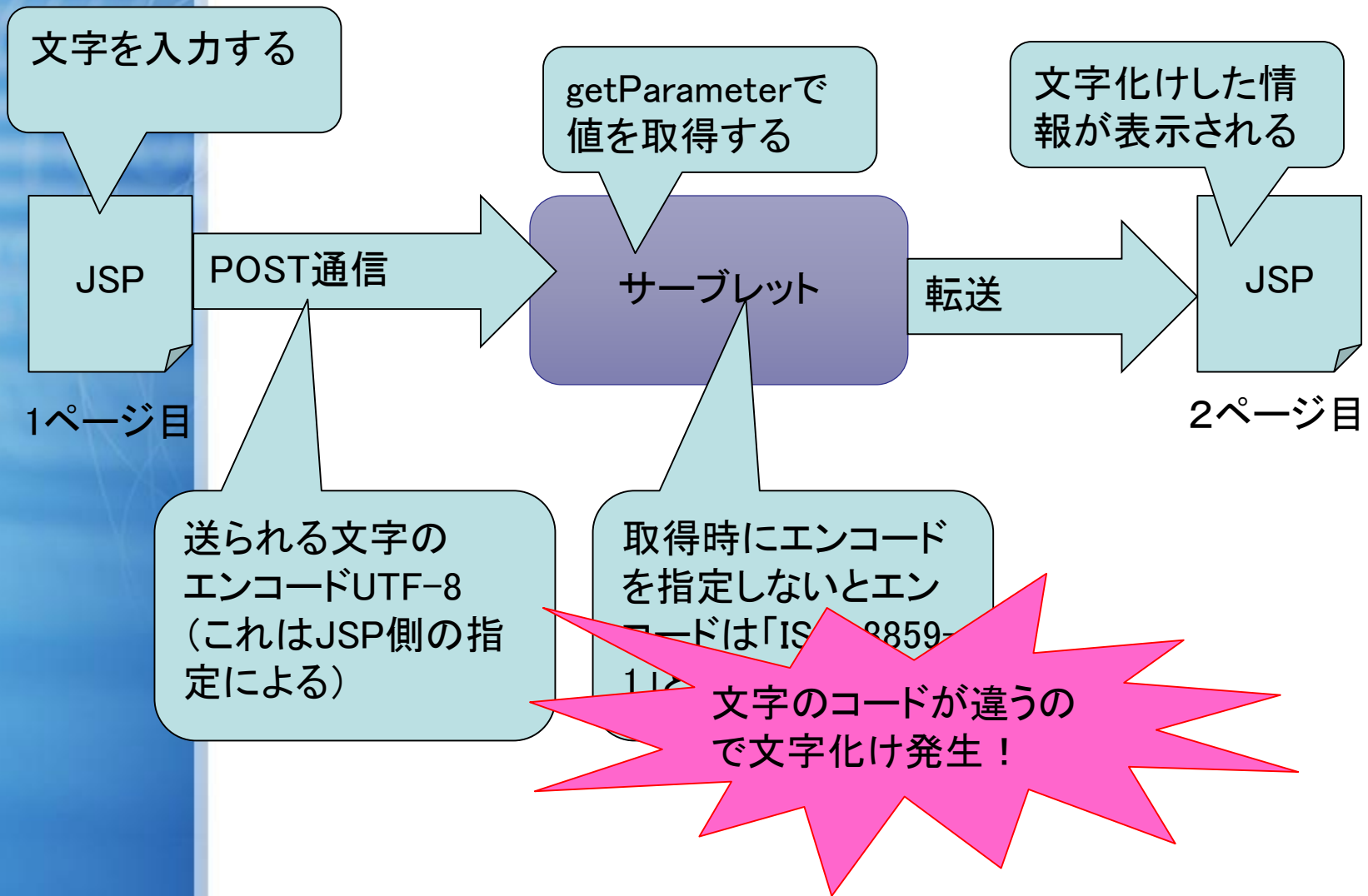
どうなった？

フィルターとは？

文字化け直ったね！

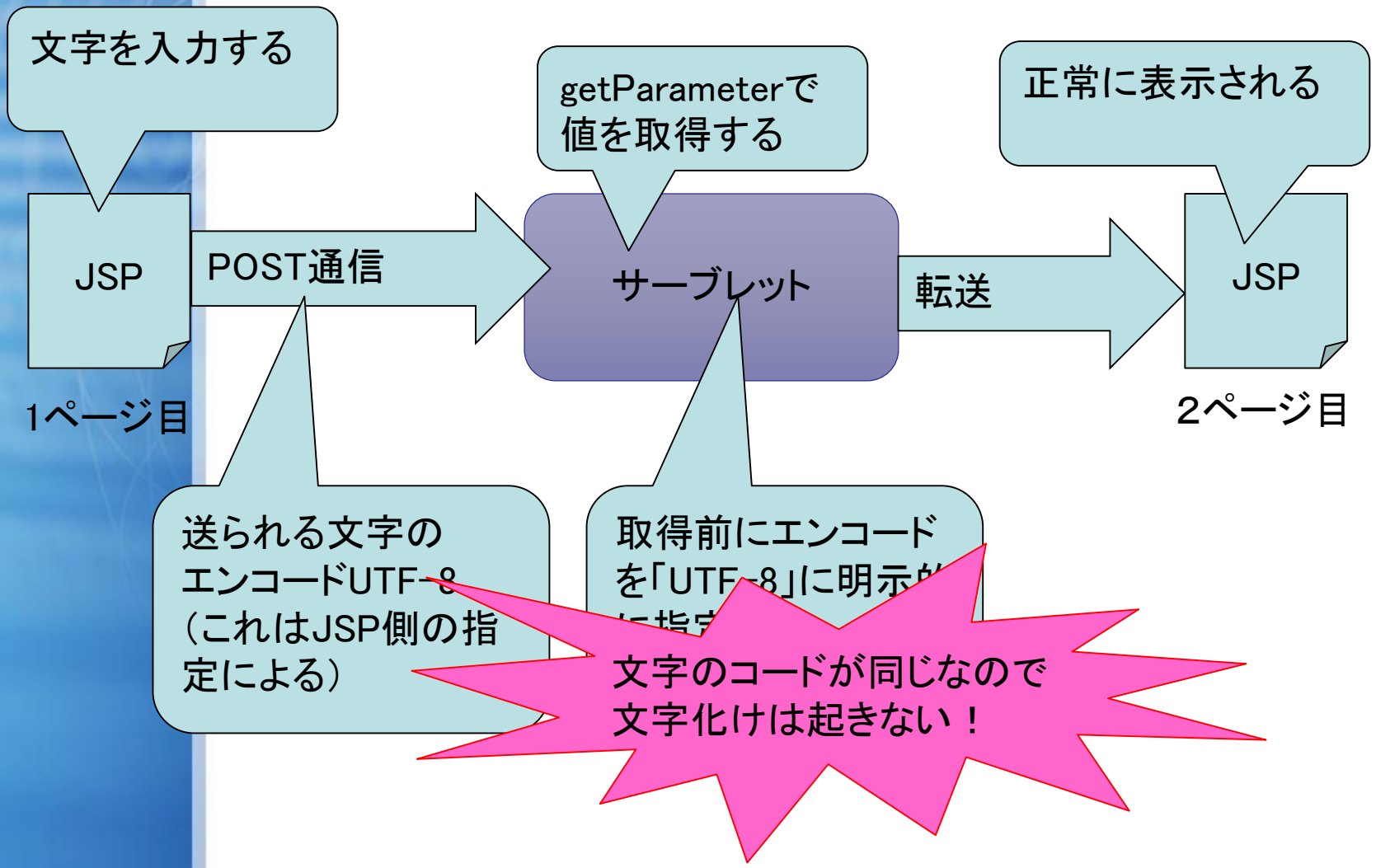
フィルターとは？

文字化けの理由



フィルターとは？

文字化けが治った理由



フィルターとは？

要するに・・・・

JSPから送られてきた
文字コードと取得するとき
の文字コードを一致させれば
文字化けは発生しない！

フィルターとは？

文字化けが治ったのは
サーブレットの頭で
request.setCharacterEncoding("UTF-8");
をやったから。

では、文字化けしない為にすべてのサーブレットの頭で
request.setCharacterEncoding("UTF-8");
をやる必要がある！

フィルターとは？

そんなのいやだー！！！

フィルターとは？

そこで登場するのが

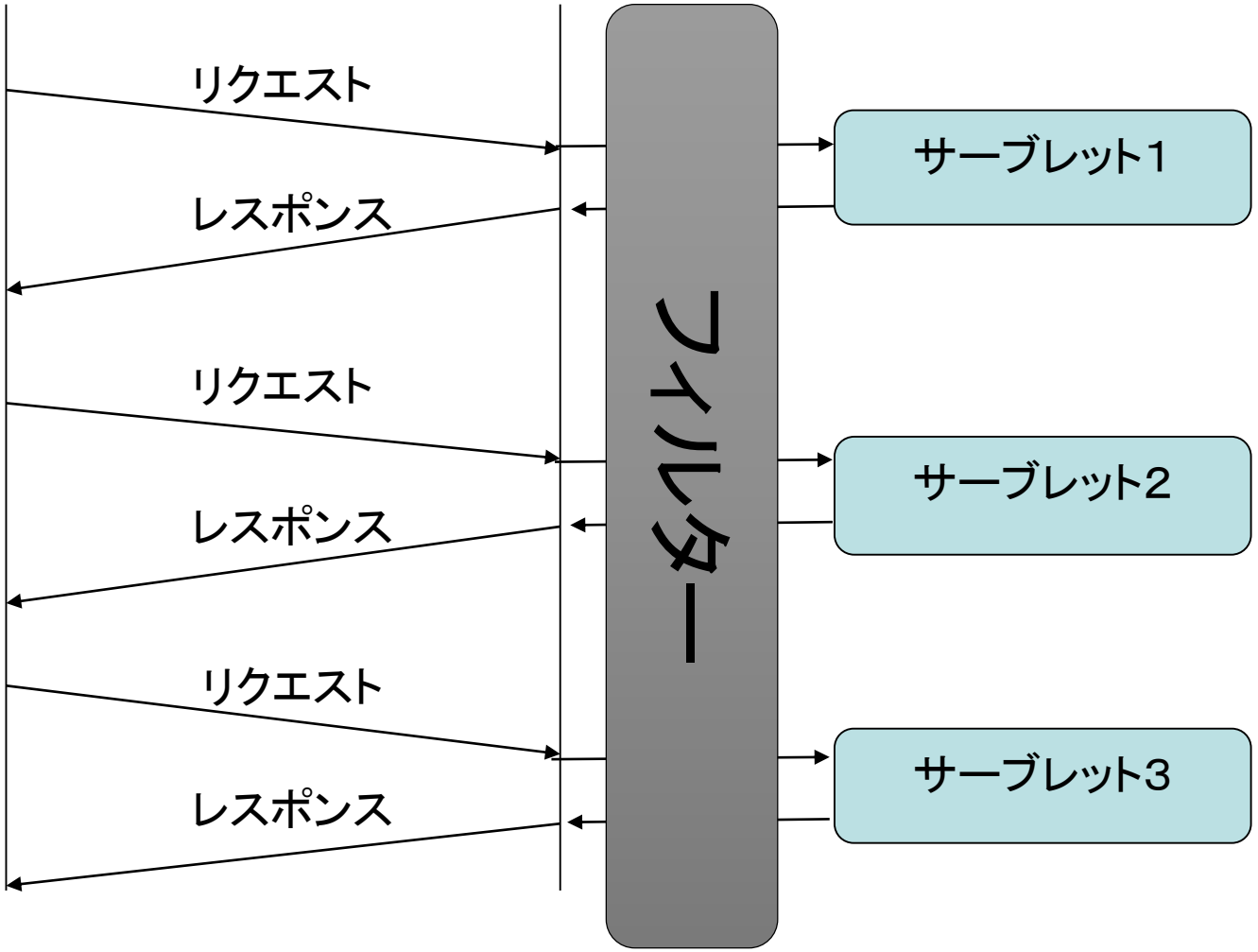
フィルター！

フィルターとは？

フィルターは、すべてのサーブレット実行の前に動作するプログラム！

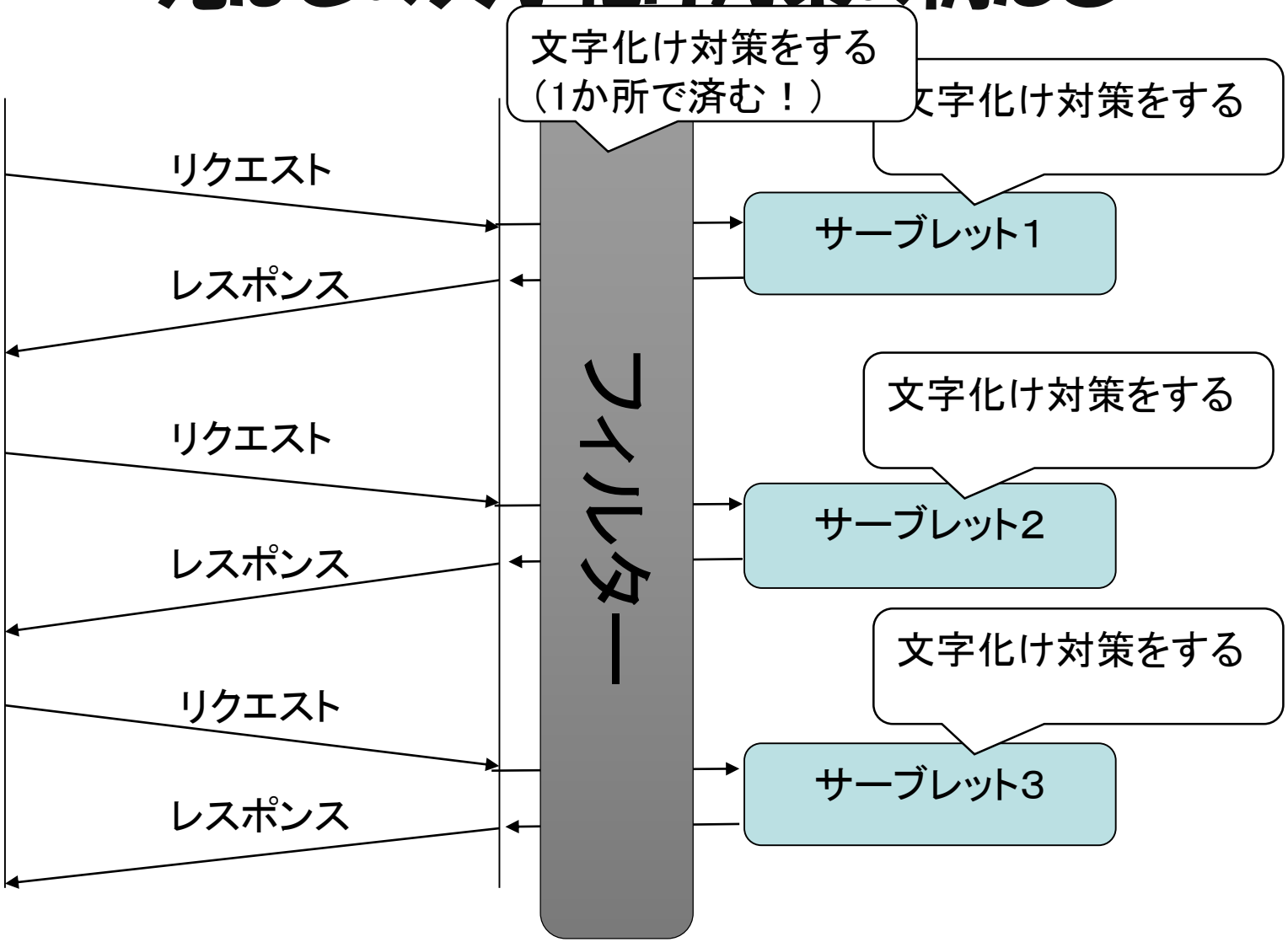
どゆこと？

フィルターとは？


リクエスト
レスポンス
リクエスト
レスポンス
リクエスト
レスポンス
フィルター
サーブレット1
サーブレット2
サーブレット3

フィルターとは？

先ほどの文字化け対策の例だと



どう実装するの？

以下の条件を満たせばOK！

- Filterクラスから派生
- @WebFilterをつける
- doFilterメソッドを実装する
- destroy、initメソッドを実装する

フィルターとは？

@WebFilterでどのURLが実行されたときに動くかを指定する

Filterインターフェースをimplementsする

```
@WebFilter("/*")
public class EncodeFilter implements Filter {

    /** エンコード */
    private final static String encoding = "UTF-8";

    @Override
    public void doFilter(ServletRequest request, ServletResponse response, FilterChain chain)
            throws IOException, ServletException {

        //日本語が文字化けしないようにする
        request.setCharacterEncoding(encoding);

        chain.doFilter(request, response);

    }

    @Override
    public void destroy() {
        ;//無処理
    }

    @Override
    public void init(FilterConfig arg0) throws ServletException {
        ;//無処理

    }
}
```

doFilterを実装する

フィルターとは？

*は「全て」なので、すべてのURLを指定したときにこのフィルターが動く

フィルターとは？

フィルターが動くときはこのdoFilterが呼び出される

```
@Override
public void doFilter(ServletRequest request, Servle
        throws IOException, ServletException {

    //日本語が文字化けしないようにする
    request.setCharacterEncoding(encoding);

    chain.doFilter(request, response);

}
```

chain.doFilterは「次のフィルター」を呼び出す処理。

フィルターとは？

Filterインターフェースは
Init、doFilter、destroyメソッドを持っている

```
public interface Filter {
    void init(FilterConfig arg0) throws ServletException;

    void doFilter(ServletRequest arg0, ServletResponse arg1, FilterChain arg2) throws IOException, S

    void destroy();
}
```

【復習】

インターフェースのメソッドはすべて「抽象メソッド」になるだったね！

要は、フィルターでは必ず init、destroy、doFilterメソッドを実装しなければならないということ！

Filterのメソッド

メソッド名	呼ばれるタイミング
init	フィルター起動時（Webアプリ起動時）
doFileter	フィルターの対象となるURLがリクエストされたとき（サーブレットが呼ばれる前）
destroy	フィルター終了時（Webアプリ終了時）

init、destroyメソッドに処理を書くことはあまりない。

目次

- フィルターとは？
- **フィルターの使用例**
- 演習

フィルターの代表的な使いかた

フィルターの使い方として、もうひとつ代表的な使い方を教えます

フィルターの代表的な使いかた

フィルターはサーブレットが実行される前に動くプログラムでした。
なので、以下のような処理をしたいときに適しています

・**複数の画面**で実施したい
　共通の処理

フィルターの代表的な使いかた

複数の画面で行いたい共通の処理って何がある？

フィルターの代表的な使いかた

一番よくあるパターンが・・・

ログインチェック

フィルターの代表的な使いかた

ログインチェックって何？

例えば・・・Amazonで

購入履歴画面

購入画面

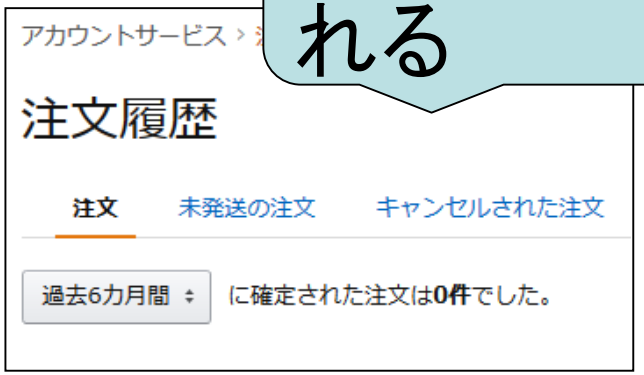
会員情報（クレジット情報など）変更

→これらの画面はログインをした後で

ないと見れないはず。

フィルターの代表的な使いかた

当然普通に見れる



②注文履歴を見る

③サインアウト(ログアウト)する

④注文履歴のURLアクセス

ログインしていないのでログイン画面に戻された

⑤ログイン画面に戻る

フィルターの代表的な使いかた

ログインしている場合

注文履歴の表示
リクエスト

ログインチェック！
結果OK！

注文履歴
サーブレット

注文履歴の表示
レスポンス

注文
履歴
画面

ログインチェック！
結果OK！

会員情報の表示
リクエスト

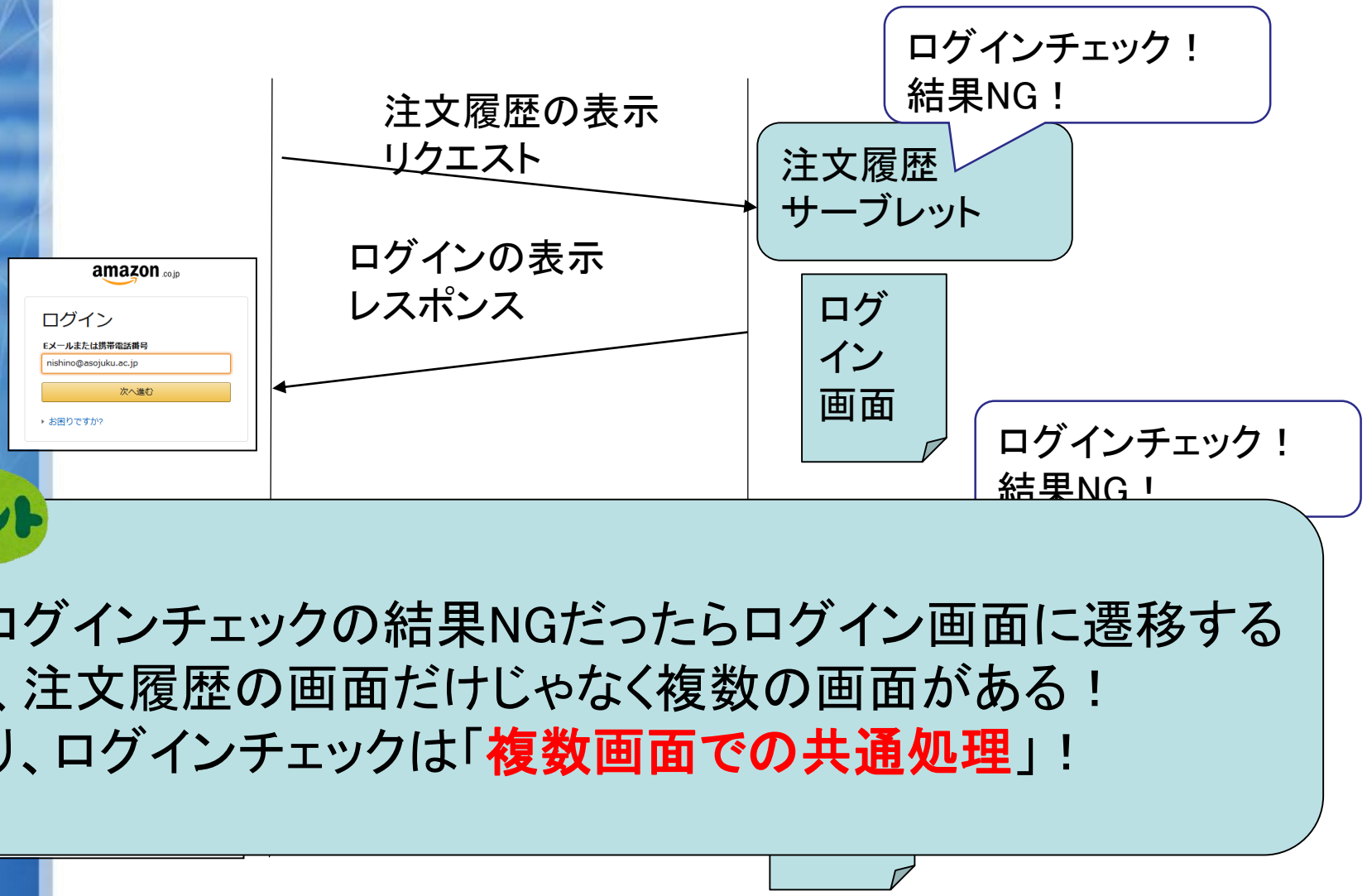
会員情報表示
サーブレット

会員情報の表示
レスポンス

会員
情報
画面

フィルターの代表的な使いかた

ログインしていない場合



ポイント

このログインチェックの結果NGだったらログイン画面に遷移するのは、注文履歴の画面だけじゃなく複数の画面がある！
つまり、ログインチェックは「**複数画面での共通処理**」！

フィルターの代表的な使いかた

フィルターを使うとこうなる（ログイン時）

注文履歴の表示
リクエスト

ログインチェック！
OKサーブレットへ処理渡す！

ログイン
チェック
フィルター

注文履歴
サーブレット

注文履歴の表示
レスポンス

注文
履歴
画面

会員情報の表示
リクエスト

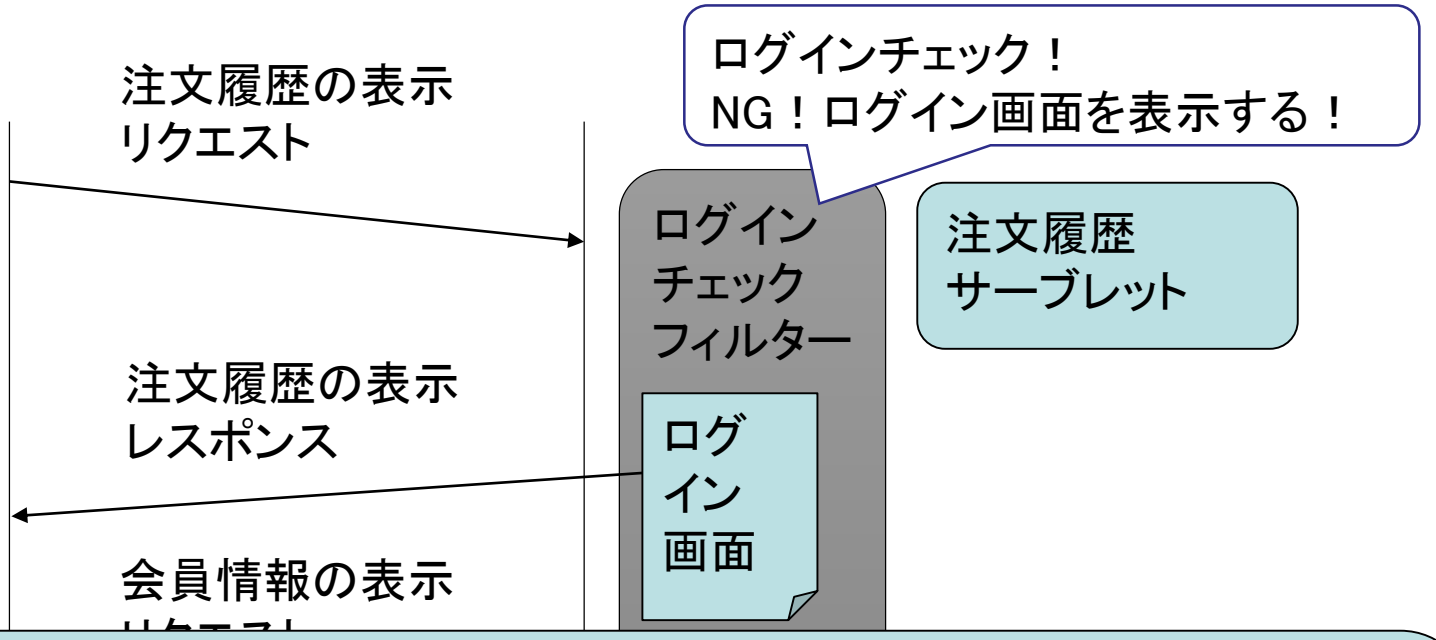
会員情報表示
サーブレット

会員情報の表示
レスポンス

会員
情報
画面

フィルターの代表的な使いかた

フィルターを使うとこうなる（未ログイン時）



ポイント

フィルターでログインNGを検知した場合、サーブレットに処理が**渡らない**！（サーブレットのプログラムは**動かない**！）

フィルターの代表的な使いかた

ここでわからないのが以下の2点だと思います。

- **ログインチェックってどうやってやるの？**
- **フィルターで画面遷移する実装方法は？**

フィルターの代表的な使いかた

まずは、

・ログインチェックってどうやってやるの？

これを見ていきたいと思います

ログインチェックの方法

ログインチェックに必要な知識

- セッション
- DB
- 簡単なJavaの知識(if文)
- MVC

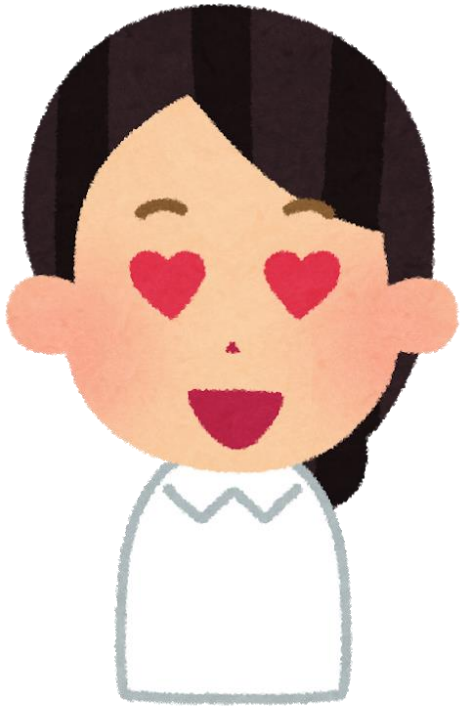
ログインチェックの方法

ログインチェックをするには

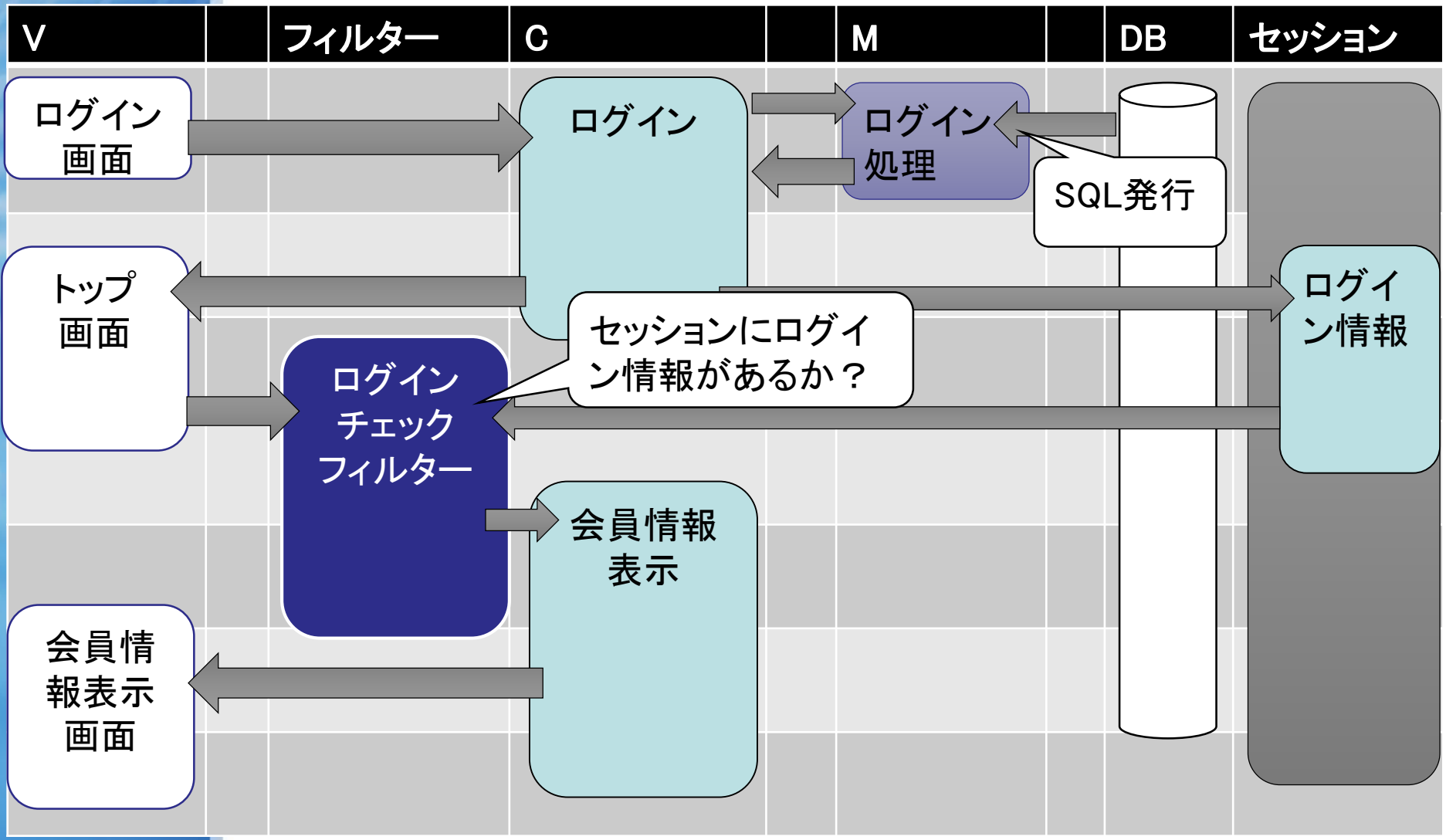
ログイン成功時に、セッションにログイン情報を保存しておき、それが存在するかどうかをチェックすればよい！

ログインチェックの方法

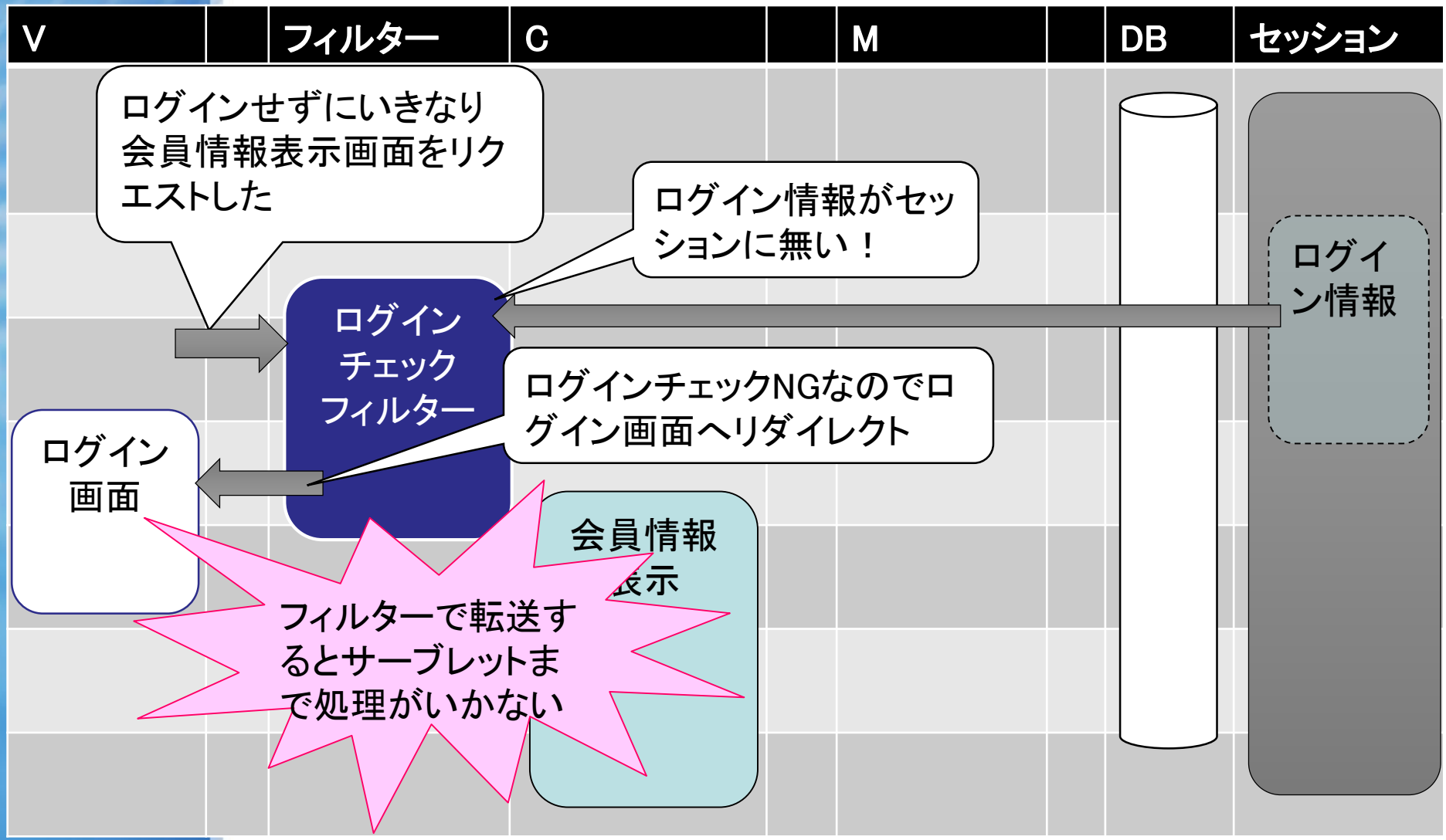
言葉で言ってもイメージが
湧かないと思うので図で表します。



ログインチェックの方法（オッケーの時）



ログインチェックの方法（NGの時）



ログインチェックの方法

つまり、以下の処理が必要！

・ログイン時にログイン情報を**セッション**に保存する

・フィルターでセッションに**ログイン情報**があるかどうかを**チェック**して、なければログイン画面へ遷移させる

次に、

- フィルターで画面遷移する実装方法は？

これを見ていきたいと思います

リダイレクト？

Filterでの画面遷移は基本的に

リダイレクト

になります。実装としては

```
//page1へリダイレクト
System.out.println("page1へリダイレクトします");
((HttpServletResponse)response).sendRedirect("page1");
```

↑は page1 にリダイレクトする記述となります

リダイレクト？

```
((HttpServletResponse)response).sendRedirect("page1");
```

sendRedirectは　HttpServletResponse クラスが持っているメソッドだが
Filterのresponse は　ServletResponseクラス。
なので、HpptServletResponseクラスにキャストして、むりやりsendRedirect
メソッドを使っている

演習

Filterを実際に動かして どのタイミングで動いたかを コンソールで確認！

photo by サニー　odai by サニー

こちらスニャーク。潜入成功。大丈夫だ。まだバレてはいない。

Webのしくみを思い出せ

https://github.com/nishino-naoyuki/2018Web/課題フィルター演習1.pdf

課題はメールでnishino@asojuku.ac.jpに送信