

自治体の港湾業務システムを
Symfony2やら
SVGやら
MapServerやら
のOSSで作ってみた。



OSC2015 Tokyo/Fall
2015年10月24日
有限会社サンビットシステム
取締役 佐々木伸幸



港湾業務システムを作った話

そもそも港湾業務って...

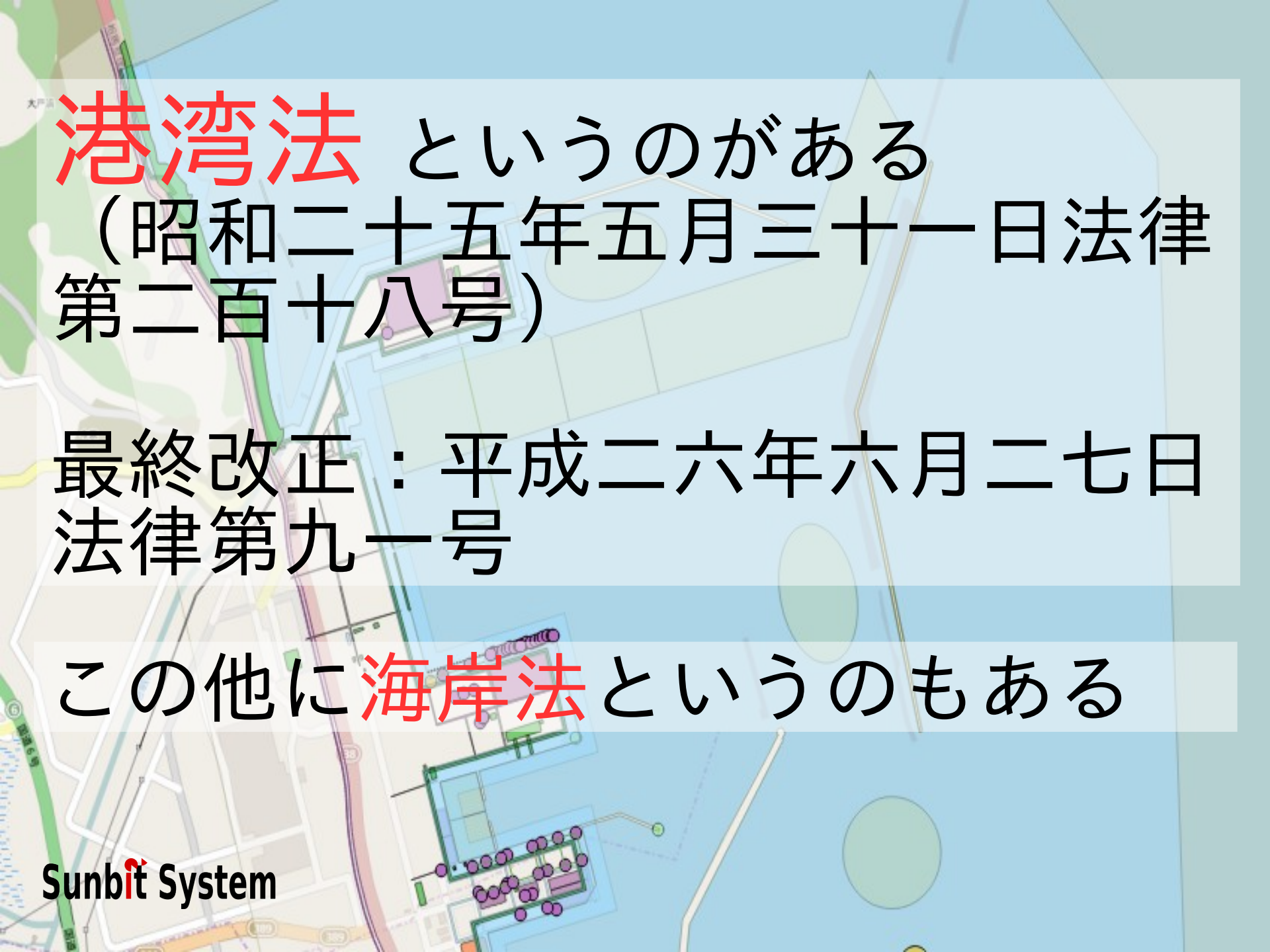
なにができたの？

Symfony2って...

SVGやらMapServerやら？

自治体の港湾業務って なんですか？

技術的「でない」話 編



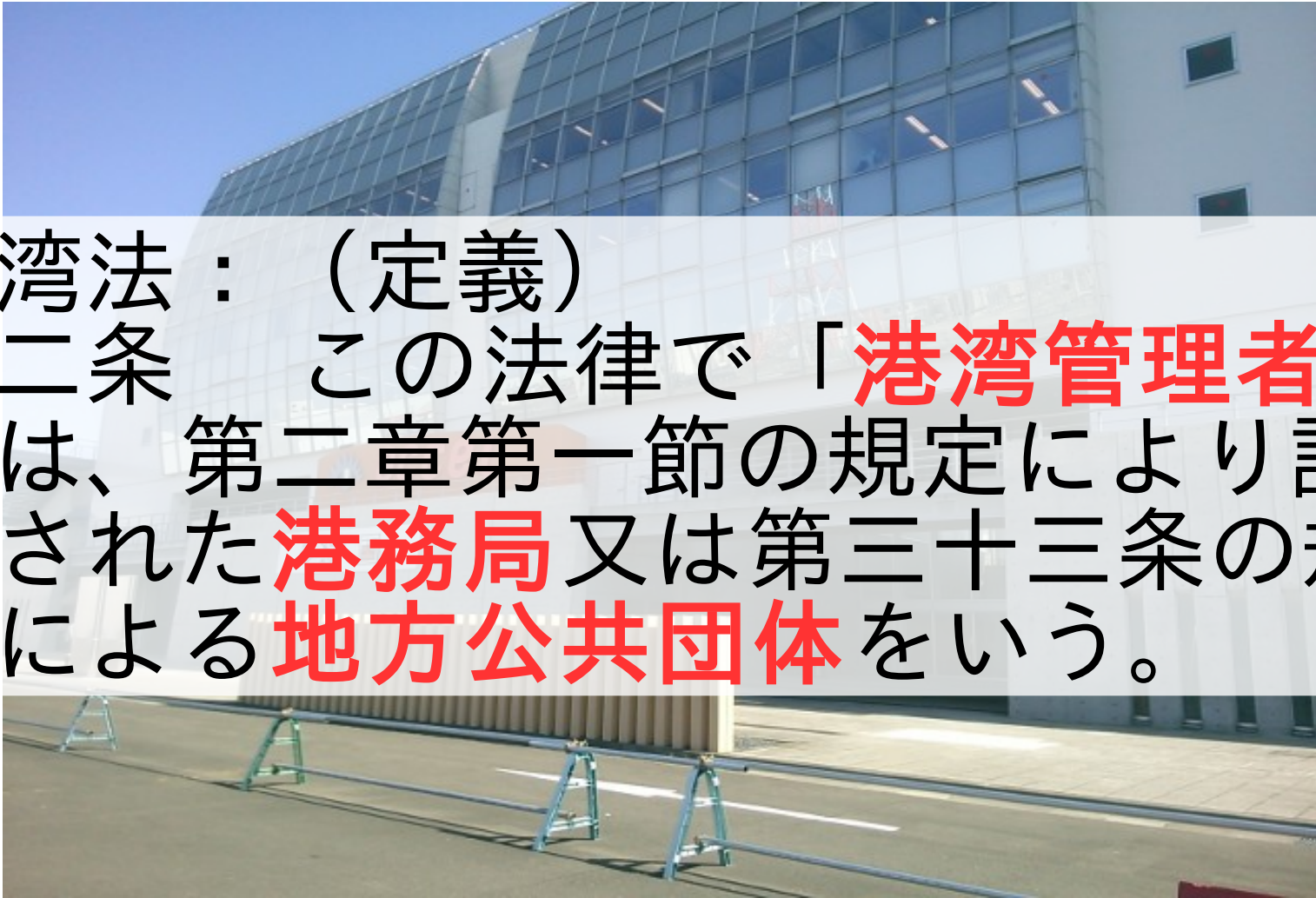
港湾法 というのがある
(昭和二十五年五月三十一日法律
第二百十八号)

最終改正：平成二六年六月二七日
法律第九一号

この他に**海岸法**というものもある

港湾法：（目的）

第一条 この法律は、交通の発達及び国土の適正な利用と均衡ある発展に資するため、環境の保全に配慮しつつ、**港湾**の秩序ある**整備**と適正な**運営**を図るとともに、**航路**を**開発**し、及び**保全**することを目的とする。



港湾法：（定義）
第二条 この法律で「**港湾管理者**」
とは、第二章第一節の規定により設
立された**港務局**又は第三十三条の規
定による**地方公共団体**をいう。

港灣法：（定義） 第二條2 港灣の 種類

(注) 東京都の洞輪沢港は避難港指定を受けているが、管理者未設立であり、かつ56条港湾ではないので本表より除く。

引用: WWW

Sunbit System

国際戦略港湾 (5港)

国際拠点港湾 (18港)

重要港湾 (102港)

www.mlit.go.jp/common/000211531.pdf

- 国際戦略港湾 (5港)
- ◎ 国際拠点港湾 (18港)
- 重要港湾 (102港)

港湾法：（定義）
第二条3 港湾の区域
「港湾区域」（水域）
「臨港地区」（陸地）

この範囲は
臨港地区

この範囲は
水域

港湾法：（定義）

第二条 5 港湾施設

水域施設

航路

外郭施設

防波堤

係留施設

臨港交通施設

航行補助施設

荷さばき施設

旅客施設...

港湾管理用移動施設

岸壁

道路

灯台

荷さばき地

港湾法：（港湾区域内の工事等の許可）

第三十七条 港湾区域内において...

一 水域（...上空及び水底の...含む。..）

又は公共空地の占用

二 ...における土砂の採取

4 港湾管理者は...許可を受けた者から
占用料又は土砂採取料を徴収することができる。

港湾法：（港湾管理者の料金）

第四十四条 港湾管理者が..施設又は役
務の利用に対し料金を徴収する場合...

第四十四条の二 港湾管理者は...入港す
る船舶から...入港料を徴収...できる。

要するに... (ここで言う港湾業務とは)

港湾管理者が

- ・ 港湾施設の利用を許可し
 - ・ 利用料を徴収する
- 業務のことです。

もちろんそれだけでなく...

2015/8月に苫小牧沖でフェリー火災が発生
原因は乗船していたリーファーコンテナを載せたシャーシの
電源部の異常と考えられる。

こうした情報の連絡調整などもしています。

港湾利用の安全管理とか...

港湾に関わる諸々について...

港湾施設の維持管理

港湾の整備計画

港湾の利用促進計画

SOLAS条約による保安対策実施

SOLAS条約：船舶の安全航行のための国際協定
タイタニック号の事故を発端に制定

港湾を適切に運営する役割

港湾法：（漁港に関する規定）
第三条 この法律は、**漁業の**用に供する
港湾として他の法律によつて指定された
港湾には適用しない。但し、当該指定さ
れた港湾で、政令で定めるものについて
は、この限りでない。

漁港は港湾ではない!!

ここまで、仕事で初めて知りました...

許可が必要な港湾施設って なんですか？



港湾管理としての主な申請許可

入出港届：船舶が港湾区域に入出する届

係留施設使用：船舶が着岸する許可

荷さばき地使用：船の積荷を保管する場所の使用許可

荷役機械使用：船舶へ荷物を揚積する機械の使用

船舶給水：船舶へ飲用水等を供給

コンテナ電源：冷凍コンテナ等に供給する電源の使用

港湾管理としての主な申請許可

物揚場：岸壁への係留と同等

船揚場：船舶を陸上保管するなど

港湾施設用地：港湾法の範囲にある土地
で整備した区画

港湾隣接地域：港湾法の範囲にある土地

海岸保全区域：海岸法の範囲にある土地

料金のかかり方（特徴的なもの）

入港料：特定の大きさの船が港湾区域に入ると課金

係留施設：船舶を係留する時間に課金

荷さばき地：使用する面積と面積単価で課金
但しペナルティ方式で
使用が長くなると高くなる

給水・電源：使用料の従量課金

料金が割引またはかからない場合

減免：連続使用など特定の条件を満たす場合に料金を割引する制度

免除：条例などによる目的を満たす場合は料金を免除する制度

国・自治体による使用：

そもそも港湾は国のもので管理を管理者に委任している。

国・他自治体が使用する場合は
申請＝>**協議**、許可＝>**同意**
に読み替える（港湾法）

An aerial photograph of a harbor scene. In the foreground, a bridge with a grey deck and blue arches spans the water. A large orange and white ship is docked at a pier, with a helicopter on its deck. The background shows a city with buildings and other boats in the harbor.

結局何やったんですか？

技術的「でない」話 編

Sunbit System

名古屋港 2014/12

港湾業務システムを作った

[トップ](#)[このサイトについて](#)[お問い合わせ](#)[ログアウト \(admin:admin \)](#)

係留施設使用許可申請

船舶入出港届作成 (Sea-NACCSと連携)

上屋・荷さばき地使用許可申請

船舶給水使用許可申請

荷役機械使用許可申請

コンテナ電源使用許可申請

SVG, 地図による視覚化

港灣施設用地使用申請詳細 表示

TOP ▶ 土地区画整理法 施設用地 申請書 協議書(案) 許可書(案) 変更申請書(案) 無償申請書 地図編集

・WEBで申請、WEBで許可、書類はPDF
一覧はCSVで取得

申請情報

[illegible]

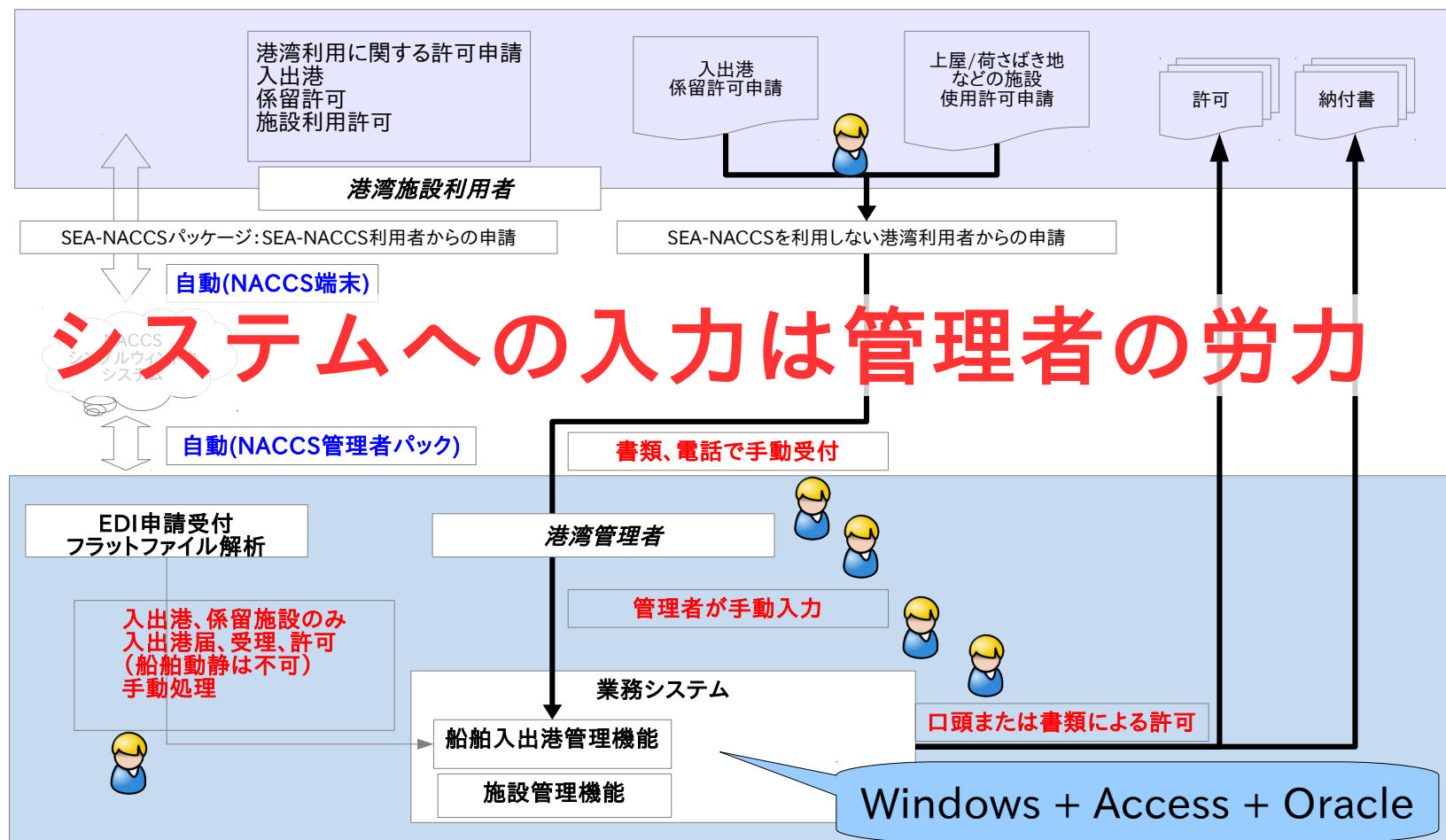
・地図はユーザが編集できる

明細情報

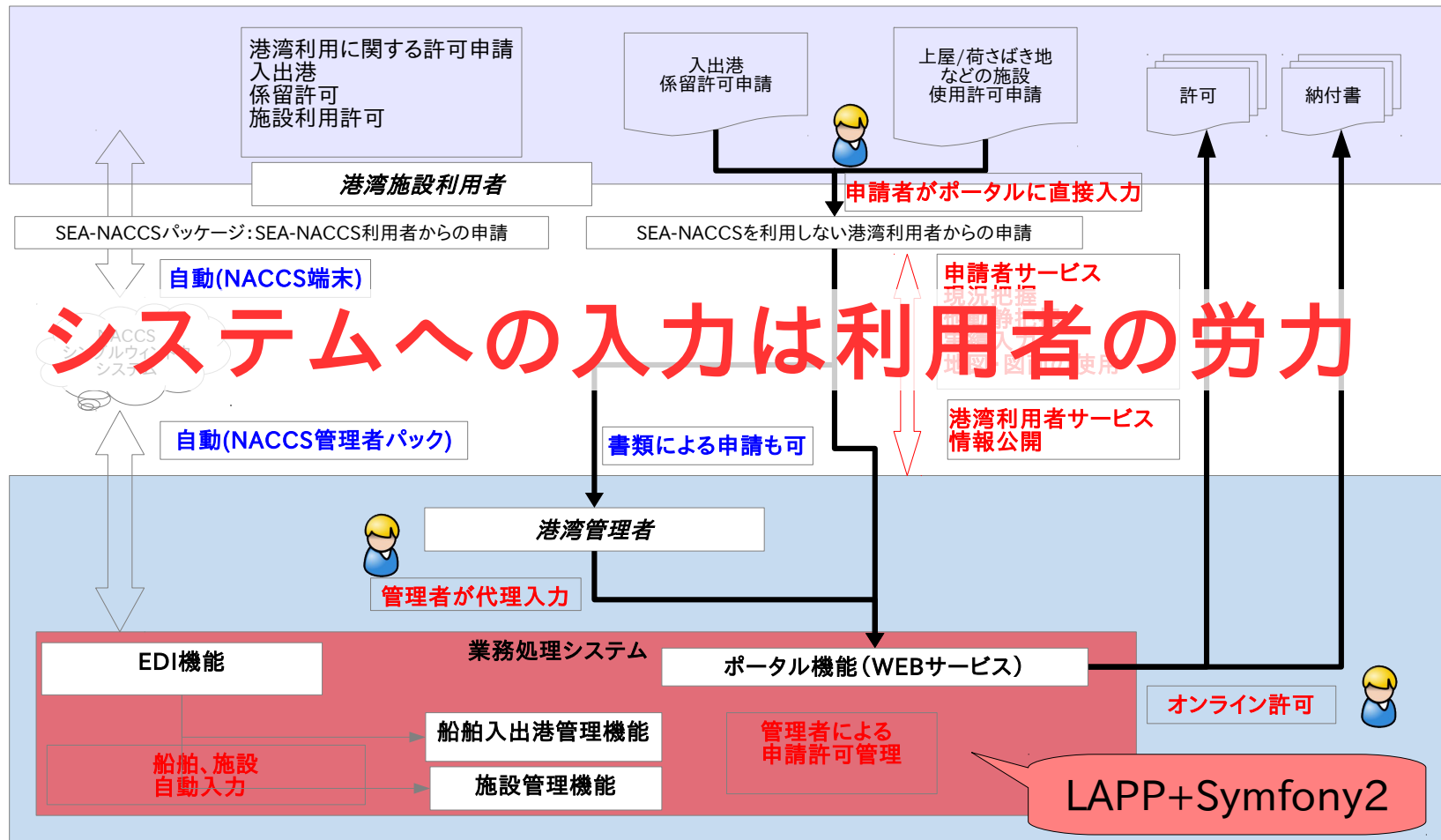
その他諸々...

使用期間	2015/04/01 - 2015/04/30		
申請区分	専用		
区画種別	利用面積	単価	小計
WX1A4 : 港湾施設用地・未舗装・4級地(西港区)	10.00 m2	30.00 円	300 円
		(減免)	0 円
料金	300円		

開発前の業務環境



企画したシステム環境



基本方針1: WEB化・オープン化

- ブラウザだけでアクセスでき、利用者側の環境整備が不要
 - 利用者個別サービス等の利便性向上
 - Linux-LTSPシンクライアントシステムと親和性が高い（この港湾管理者では既に導入済み）
 - Linux、オープンソースを使用することで、ベンダロック排除、ライセンスコスト削減が見込める
-
- 成果はオープンソースにしたい
 - 現在オープンソース部分の公開作業中

基本方針2: 視覚化・自律化

- 電子地図・情報図等での視覚化によるリアルタイム状況把握
利用者相互調整の自律化
(管理者による調整削減)
- 情報公開 (公開にかかる手数の削減)
例: 新聞の「出船入船」への情報提供をこれまで港湾管理者がFAXしていた
→WEBで利用者が取得するので
管理者の費用発生なし

基本方針2: 視覚化・自律化

全選択してから一つずつ消すと面積が残るなど、操作できない小さなブロックがある場合ブラウザの表示拡大をして確認してください。
Ctrlと'+'キーで拡大、Ctrlと'-'キーで縮小、Ctrlと'O'でリセットできます。(ブラウザにより違いがあります)

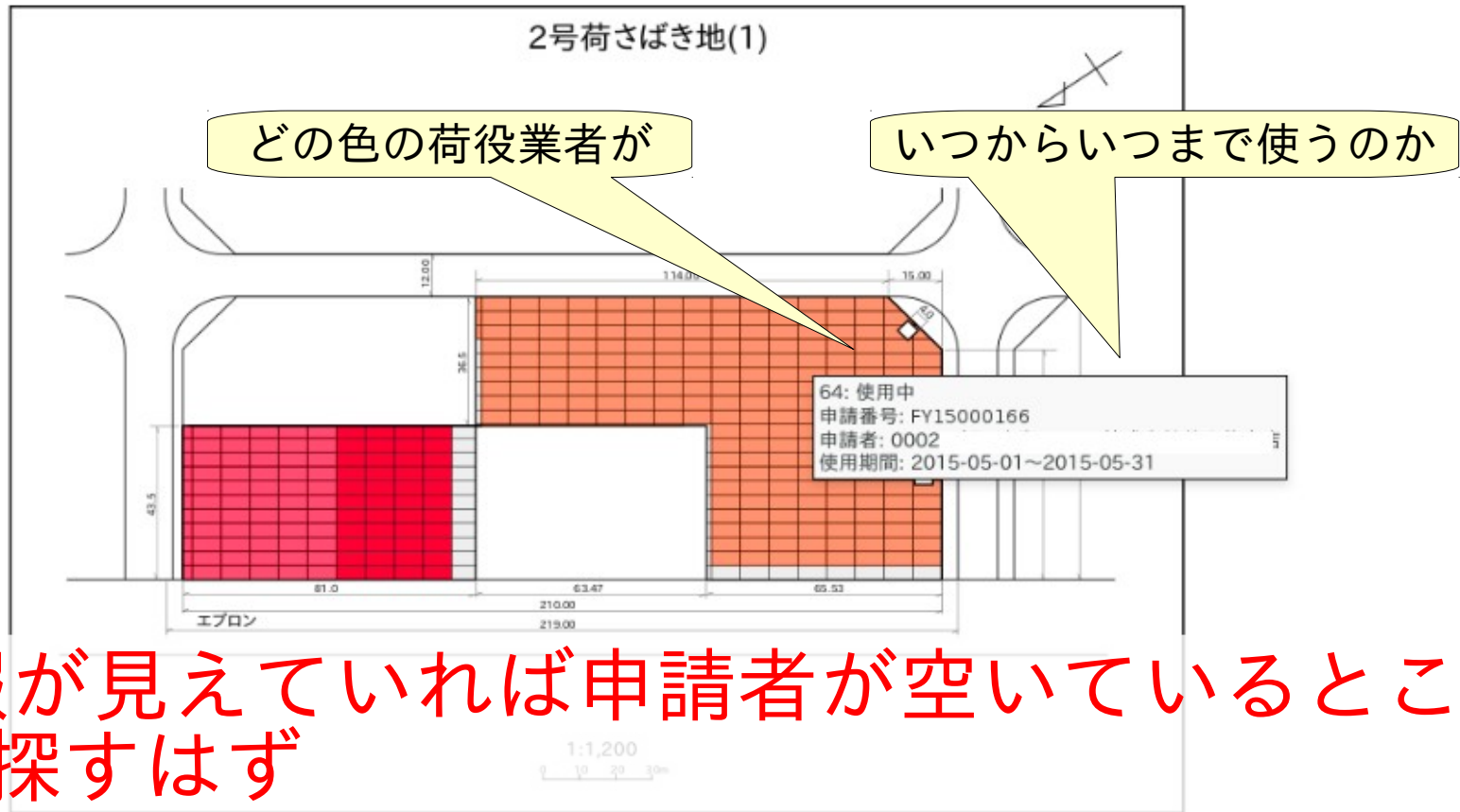
確定

やりなおし

ブロック全選択

閉じる

使用面積:1360.4 m2



- 情報が見えていれば申請者が空いているところを探すはず

基本方針2: 視覚化・自律化

TOP 船舶入出港管理 船舶入出港一覧

▼表示条件

公共在港船一覧 会議対象予定船一覧(05/22) 仮バース船一覧 エラー一覧 東港防波堤通過一覧

視察用一覧 新聞社用一覧(入港) 新聞社用一覧(出港) 私設岸壁係留一覧 入/出港届未提出一覧

最近 取消された申請

代理店コードを入力して検索ボタンをクリックすると、指定された代理店の料金確認前のデータが表示されます。受付番号を入力して検索した場合はデータの状態によらず指定された受付番号のデータが表示されます

どの代理店が いつからいつまで どの岸壁に船を係留するか

詳細な検索条件と指定

新規申請 PD CSV出力

受付番号	代理店名 略称	船名	全長	総トン数 みなしトン数	入港 喫水	入港日時	着岸日時	離岸日時	バース ビット	仕出港 仕向港	貨物
① WV15004607-1		ヒーシエ8	98.00	2,983.00 0.00	5.60	(*)05/16 01:45	(*)05/19 06:16	05/21 06:00	中央南2 18 ~ 23	チンファンタオ(秦皇 島) カオシュン(高雄)	揚・植 料

- 他者が見えると申請者間で調整できるはず
＞管理者だけが申請情報を持つとできない
休日にも調整当番は不要

実現する方法

ベースシステム

LAPP+Symfony2

係船図等の生成

CANVAS

荷さばき地等区画

SVG

地図表示

WMS

地図描画

WFS

地図制作

ESRI:shp

全部が標準技術の上に載るように。
HTML5でいい。IEの古いのとかナシ。

システム開発の行程

2012/10	ヒアリング
2013/02	プロポーザルに応募
2013/06-2014/03	1次開発(船舶関連完成)
2014/06-2015/03	2次開発(全完成)
2015/04	運用開始

OSS環境での業務システム開発

プロポーザル = 提案主軸は楽しい
実現に至るまでの過程で、OSSを利用
することのメリットを理解してもら
うのは、割と大変

OSSオンリー = 自由な組合せが楽しい
→ 解決できないときは苦しい。
情報を読み取る力が必要。
完成品はサポートできるようになる



LAPPってなんですか？

Home ニュース 特集 Linux コミュニティ **OSSナビ** リソース ビデオ

Home > OSSナビ > OSSナビ

OSSナビ

Open Source Software Navigation



Linux/OSSの歩き方

どんなことでも、誰も新しい文化にはなじめないものです。

Linux/OSSも、例外ではありません。日々進歩、変化する多様な情報を使いこなし、活用していくには、いくつかの前提やちょっとしたコツが必要です。

事前にコツを知ること、理解が早まり、安全で効率的に活用できると考え、いわばコツといえるようなLinux/OSS技術者の常識について、Windows経験3年目の技術者を想定読者と仮定し、対話形式でわかりやすい解説をしています。

<http://www.linuxfoundation.jp/content/linux-oss-track-j>

技術的な話 編

Windows技術者のためのLinuxの概念と操作

Windowsを良く知っている技術者でもLinuxを初めて使おうとすると、その概念の違いや操作性の違いに戸惑うこ

LINUX
FOUNDATION

Membership

Join the Linux Foundation

and support the organization that employs Linus Torvalds and enhances, promotes and protects Linux.

メンバーとしてLinuxの進歩に携わりませんか？

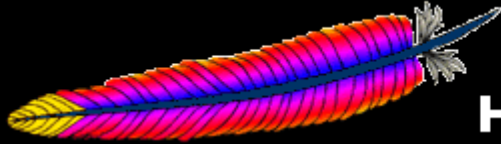
メンバーシップ詳細 >>

Linux - Where Are We Going

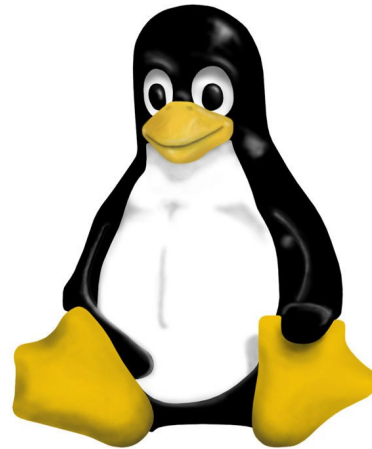


VIDEOS





HTTP SERVER PROJECT



以上



a SensioLabs Product



Symfony2ってなんですか？

*Symfony is a set of reusable
PHP components...*

The standard foundation on which the best PHP applications are built. Choose any of the 30 stand-alone components available for your own applications.

[Browse components →](#)



*... and a PHP framework for
web projects*

Speed up the creation and maintenance of your PHP web applications. End repetitive coding tasks and enjoy the power of controlling your code.

[What is Symfony →](#)

Projects using Symfony



SensioLabs

日本語

Powered by Google 翻訳

Créateur de  Symfony

Symfony2とは

一目でSensioLabs

PHPによるWEBアプリケーションの フレームワーク (SensioLabs:フランス) 特徴

「バンドル」という単位が基本セット。
制作するアプリケーションはバンドルと
して作る。(Symfony2自身もバンドル)

ORMは標準でDoctrineを含む。
Symfony2とは独立して開発

Sunbit System

SensioLabsインターナショナル

92-98 大通りヴィクトル・ユーゴー
92-98 大通りヴィクトル・ユーゴー
フランス contact@sensiolabs.com

SensioLabsインターナショナル

92-98 大通りヴィクトル・ユーゴー
92-98 大通りヴィクトル・ユーゴー
フランス contact@sensiolabs.com

SensioLabsインターナショナル

Neusser-Farner
60670 Frankfurt
ドイツ
電話: +49 (0) 221 66 99 27 50
ファックス: +49 (0) 221 66 99 27 79
mail@sensiolabs.de

SensioLabsイギリス

キャッスルウッドハウス
77-91 ニューオックスフォード・ストリート
ロンドン WC1A 1DG
イギリス
電話: +44 20 3179 9555
info@sensiolabs.co.uk

Symfony2 特徴（つづき）

twigテンプレートエンジンを含む。
Symfony2とは独立して開発

Symfony2の個人的使用感

- それなりに大きなフレームワーク。Maple+Smartyなどの手軽さはない。
- 手軽ではないがSymfonyバンドルそのものが参考書になり、コーディング規約等はある程度統一されるので大規模開発には向いてる気がする。
- twig賢い。
- 日本Symfonyユーザ会に感謝（ドキュメント訳）
<http://docs.symfony.gr.jp/>

Symfony2

おおまかな開発の流れ

- ・ **Bundle**を作る。
- ・ データ設計に従い**Entity**を書く。
(入力バリデータも、ORMもここ) →anntation大切
- ・ データ設計に基づき**Repository**を書く。
- ・ 画面設計に従い**Form**を書く
- ・ 画面設計に従い**twig**テンプレートを書く
- ・ 動作設計に従いルーティングとアクション(**Controller**)を書く
- ・ 必要に応じてサービスを書く

Bundle:まず最初に

アプリケーションの枠組みを定義する

コマンド一発

```
$ php app/console generate:bundle  
  --namespace=Watatsumi/FacilityBundle  
  --format=yml
```

注意
バンドルに限らず、コントローラ、フォーム、エンティティ等は命名規則
がわりと厳格で、規則を無視すると痛い目に会います。

Entity:データを定義する

変数を定義する

DBにマップする場合はORMでカラム名を定義する(annotation)

```
Class DFacBase
{
    /**
     * @var integer
     *
     * @ORM\Column(name="id", type="integer", nullable=false)
     * @ORM\Id
     * @ORM\GeneratedValue(strategy="SEQUENCE")
     * @ORM\SequenceGenerator(sequenceName="d_cargo_id_seq",
allocationSize=1, initialValue=1)
     */
    protected $id
}
```

Entity:データを定義する

フォームと結びつける場合は`validator`を定義すると入力チェックできる

```
Class DFacBase
{
    /**
     * @var float
     *
     * @ORM\Column(name="area_size", type="decimal", nullable=true)
     * @MstAssert\NotBlank(
     *     groups={"dcargo_add"}
     * )
     * @MstAssert\Numeric(
     *     precision = 2,
     *     min=0.01,
     *     max=999999.99,
     *     groups={"dcargo_add"}
     * )
     */
    protected $areaSize;
}
```

Entity:データを定義する

getter, setterでデータを取得、設定できるようにする

```
Class DFacBase
{
    public function setAreaSize($areaSize)
    {
        $this->areaSize = $areaSize;

        return $this;
    }

    public function getAreaSize()
    {
        return $this->areaSize;
    }
}
```

Entity:データを定義する

Doctrineには便利メソッドがありオーバーライドできる

Class DFacBase

```
{
    /*
     * persist()を発行する前に自動で動いてくれるメソッド
     * 更新日時を自動で設定するなど使える
     */
    public function prePersist()
    {
        $this->createAt = new \DateTime();
        $this->updateAt = new \DateTime();
        if($this->getUseTo() instanceof \DateTime) {
            $this->setUseTo($this->getUseTo()->setTime(23, 59, 59));
        }
    }
}
```

XXXRepository.phpを書く

DBからEntityへデータを読み込むメソッドを書く

find()等は定義済み。idをシーケンスにするならクラス定義だけでもよい

```
class DCargoRepository extends EntityRepository
{
    public function searchOneByFacCode($facCode, $facType=null)
    {
        $qb = $this->createQueryBuilder('p')
            ->where('p.delFlg = :delFlg')->setParameter('delFlg', 'FALSE')
            ->andWhere('p.facCode = :facCode')
            ->setParameter('facCode', strtoupper($facCode));
        $qb = $qb->andWhere('p.facType in(:facType)')
            ->setParameter('facType', $facType);
        try {
            $mfaccargo = $qb->getQuery()->getOneOrNullResult();
        } catch (\Exception $e) {
            $mfaccargo = null;
        }
        return $mfaccargo;
    }
}
```

Form:画面入力値を定義する

HTML上に展開される変数。画面とプログラムのやりとりに使う。

Entityと同じ変数はEntityに自動的にマップされる。

EntityにマップされるとEntityに書いたvalidataが働く

```
class DCargoAddType extends AbstractType
{
    public function buildForm(FormBuilderInterface $builder, array
$options)
    {
        $builder
            ->add('areaSize', 'text', array('required'=>true, 'max_length'=>9,
'attr'=>array('size'=>11)))
            ->add('useFrom', 'date_text', array('required'=>true,
'format'=>"yyyyMMdd", 'max_length'=>8, 'attr'=>array('size'=>10)))
            ->add('useTo', 'date_text', array('required'=>true,
'format'=>"yyyyMMdd", 'max_length'=>8, 'attr'=>array('size'=>10)))
    }
}
```

twig:画面を定義する

twig変数や条件判断を利用してHTMLを生成する。

```
{% if dcargo.mainSts == constant('DCargo::MAIN_STS_ERROR') %}  
<tr>  
  <td bgcolor="yellow">  
    <font size="+2">エラー理由</font> <br />  
    {{ dcargo.delReason|nl2br }}  
  </td>  
</tr>  
{% endif %}  
<tr>  
  <th><label>使用面積<em>*</em>:</label></th>  
  <td colspan="3">  
    {{ form_widget(form.areaSize) }}  
    <button type="button" id="searchMap" click="openBlk()">  
      ブロック図表示</button>  
    <div class="valid-error">{{ form_errors(form.areaSize) }}</div>  
  </td>  
</tr>
```

input属性へ展開

validate時のエラー表示

routing.yml:コントローラを定義する

ルーティング: URLに対するControllerのメソッド（アクション）を定義

```
watatsumi_facility_cargo_add:
  pattern    : /cargo/add
  defaults   : { _controller: WatatsumiFacilityBundle:DCargoAdd:add }
  requirements:
    _method: GET|POST

watatsumi_facility_cargo_edit:
  pattern    : /cargo/edit/{id}
  defaults   : { _controller: WatatsumiFacilityBundle:DCargoAdd:edit }
  requirements:
    _method: GET|POST

watatsumi_facility_bundle_controller_dcargo_add_controller_phpのeditアクション
  pattern    : /cargo/confirm
  defaults   : { _controller:
WatatsumiFacilityBundle:DCargoAdd:confirm }
  requirements:
    _method: POST
```

XXXController.phpを書く

本筋のコード書き：ここまで来ればこれを書く と動く

Controllerにビジネスロジックを書く|書かないはプロジェクトポリシー

```
class DCargoAddController extends Controller
{
    /**
     * \brief 登録画面表示処理
     * \param なし
     * \return dcargoadd.html.twigの画面情報
     * \details 許可データを新規生成し、登録画面を表示する
     */
    public function editAction()
    {
        $request = $this->getRequest();
        $dcargo = new DCargo();
        $dcargo->setRegistAt(new \DateTime('now'));
        $dcargoFix = new DCargoFix();
        $dcargoFix->addCargo(new DCargoCargo());
        $dcargo->addFix($dcargoFix);
    }
}
```

routingで定義されたアクションは
xxxxActionとなる命名規則がある

XXXController.phpを書く

FormとEntityはどうやって結びつくか

```
// POSTなどのリクエスト情報
$request = $this->getRequest();
$id = $request->get('id');

// DBからEntityの生成
$dcargo = $this->getDoctrine()
    ->getRepository('Watatsumi\FacilityBundle:DCargo')
    ->searchOneById($id);

// DBから取得したEntityと記述したFormを結びつける
$form = $this->createForm(new DCargoAddType(), $dcargo);

// POSTされた内容をEntityに定義されたValidataで評価する
$form->bind($request);
if(!$form->isValid()) {
    // errorを設定し元画面を呼び出す
}
```

Form

Entity

2.3はこれ
2.7では変わった

form_error()に表示される

他の技術ネタ



a SensioLabs Product



- What is Symfony?
- Get started
- Documentation
- Community
- Marketplace
- Jobs
- Business Solutions
- News

*Symfony is a set of reusable
PHP components...*

The standard foundation on which the best PHP applications are built. Choose any of the 30 stand-alone components available for your own applications.

[Browse components →](#)

CANVAS ?



*... and a PHP framework for
web projects*

Speed up the creation and maintenance of your PHP web applications. End repetitive coding tasks and enjoy the power of controlling your code.

[What is Symfony →](#)

Projects using Symfony



CANVAS: ブラウザ単独でグラフィック

ブラウザに自由に絵が書ける(けい船図に使用)

西港区 東ふ頭・入船ふ頭 係船図 (2015/03/19 00:00 ~ 2015/03/19 14:00)

凡例: 着岸済 = 緑, 許可済 = 青, 未許可 = 黄, 優先区分船(許可状態によらず) = 赤

	EH06C						EH05C						EH04C					
	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76
03/19 00:00																		
03/19 01:00														ひだか[VV15001926] 03/19 01:00-03/19 20:30				
03/19 02:00																		
03/19 03:00																		
03/19 04:00																		
03/19 05:00	第十五福神丸[VV15002940] 03/19 05:35-03/19 08:40																	
03/19 06:00																		
03/19 07:00							扇鳳丸[VV15002887] 03/19 07:30-03/19 16:00											
03/19 08:00																		
03/19 09:00	祐徳丸[VV15003024] 03/19 09:15-03/19 18:00																	
03/19 10:00																		
03/19 11:00																		
03/19 12:00																		
03/19 13:00																		

シンプルに矩形位置計算しfillの例

```
var ctx = moorcanvas.getContext('2d');  
// 外枠とグラフ描画エリアの枠を描く。  
ctx.beginPath();  
ctx.strokeStyle = "rgb(0, 0, 0)";  
ctx.fillStyle = "rgb(248, 248, 248)";  
ctx.fillRect(lArea, tArea, wArea, hArea);  
ctx.strokeRect(lArea, tArea, wArea, hArea);  
ctx.closePath();  
ctx.stroke();
```

```
...  
// Vesselの矩形を描画する。  
ctx.beginPath();  
switch (vessel[viColor]) {  
  case 0: // 許可済み = 青  
    ctx.fillStyle = 'rgba(131, 202, 255, 0.7)';  
    break;  
  case 1: // 未許可 = 黄色  
    ctx.fillStyle = 'rgba(256, 256, 0, 0.7)';  
    break;  
  、  
  ....
```

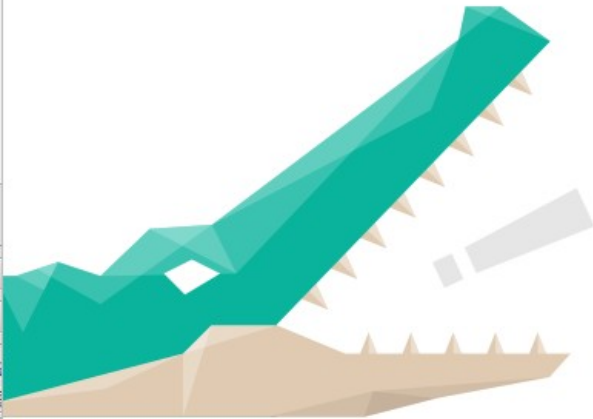
opacity
重複すると色も重なる

Snap.svg

The JavaScript SVG library for the modern web.



SVG ?



Sunbit System

Why SVG (and Snap)?

SVG is an excellent way to create interactive, resolution-independent vector graphics that will look great on any size screen. And the Snap.svg JavaScript library makes working with your SVG assets as easy as jQuery makes working with the DOM.

Modern features for modern browsers

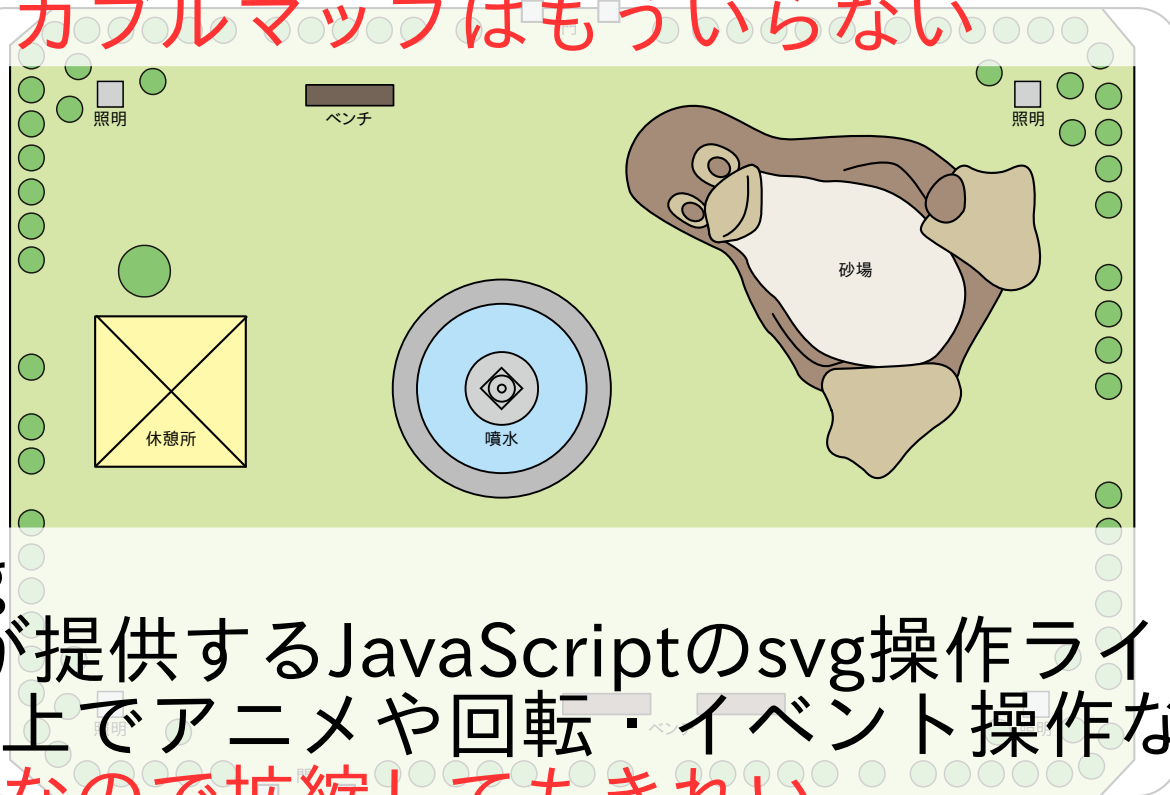
Snap.svg is designed for modern browsers and therefore supports the newest SVG features like masking, clipping, patterns, full gradients, groups, and more.

SVG: Scalable Vector Graphic

ペンギン公園

図をマウスで選択したり、ホバーすると情報表示などが行える

→ クリックابلマップはもういらない



Snap.svg

ADOBEが提供するJavaScriptのsvg操作ライブラリ
ブラウザ上でアニメや回転・イベント操作など可能

→ ベクタなので拡大縮小してもきれい

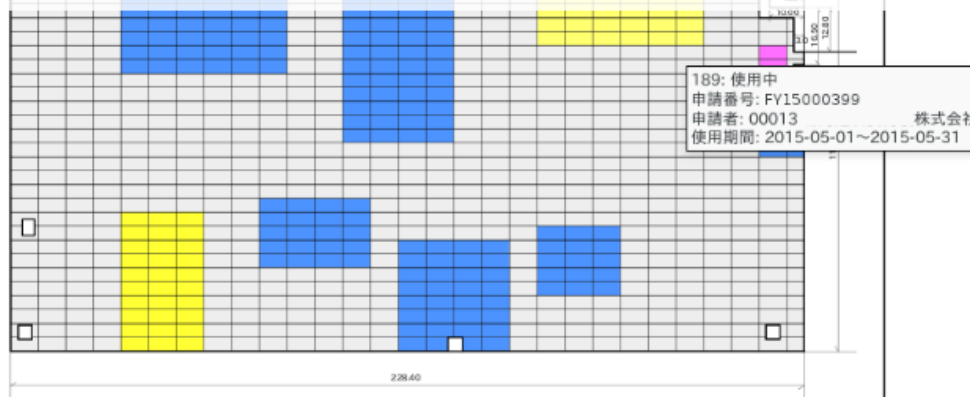
SVG: ブラウザが直接解釈可能なXML

```
<article class="module width_full">
  <header><h3>ブロック図</h3></header>
  <object id="svg"> <svg version="1.1" width="841.891px"
height="595.279px" xmlns="http://www.w3.org/2000/svg"
xmlns:svg="http://www.w3.org/2000/svg"
xmlns:xlink="http://www.w3.org/1999/xlink">
  <g id="グリット01_1_">
  <g id="グリット01_2_">
    <rect id="490" x="220.881" y="324.163" fill="#eeeeee"
width="22.531" height="11.326">
      <desc>block:490, size:32, road:false, disable:false</desc>
      <title>490: 使用可</title>
    </rect>
    <rect id="519" x="220.881" y="335.49" fill="#eeeeee"
width="22.531" height="11.327">
      <desc>block:519, size:32, road:false, disable:false</desc>
      <title>519: 使用可</title>
    </rect>
  </svg> </object></article>
```

SVG=WEB力が高い

このシステムでの例：

- 1) 利用者色を決めておく
- 2) 現在のデータから使用中領域を色付けする
- 3) 色付けしないところは選択イベントを設定する
→ イベントがないところは選択できない
→ 他者が使用しているブロックは選択できない



人手で重複チェックをして区画許可
→ 重複チェックは不要（空きしか選べない）



Navigation

[About](#)
[Products](#)
[Community](#)
[Development](#)
[Download](#)
[Documentation](#)
[FAQ](#)



MapServer is a project of the [Open Source Geospatial Foundation](#).



Open HUB \$3.41M Cost

クイック検索

検索

モジュール、クラス、または関数名を入力してください

Welcome to MapServer

MapServer is an [Open Source](#) platform for publishing spatial data and interactive mapping applications to the web. Originally developed in the mid-1990's at the University of Minnesota, MapServer is released under an [MIT-style license](#), and runs on all major platforms ([Windows](#), [Linux](#), [Mac OS X](#)). MapServer is not a full-featured GIS system, nor is it as easy to use.

The MapServer [Project Steering Committee \(PSC\)](#), sanctioned by [OSGeo](#), manages and administers the project which is maintained, improved, and supported by developers and users from all over the world. See the [community activities](#) page for mailing lists, etc.

[More about MapServer](#)

MapServer Suite Home

[MapServer](#), [MapCache](#), [TinyOWS](#) home

Download the MapServer Suite

[Download the suite](#)

Recent Announcements

2015-02-12 - MapServer 7.0.0-beta1 is released

The first beta for MapServer 7.0 has been released. The list of major changes for this release can be found in the [7.0 announcement](#). See the [changelog](#) for the full list of changes and head to [Download](#) to obtain a copy.

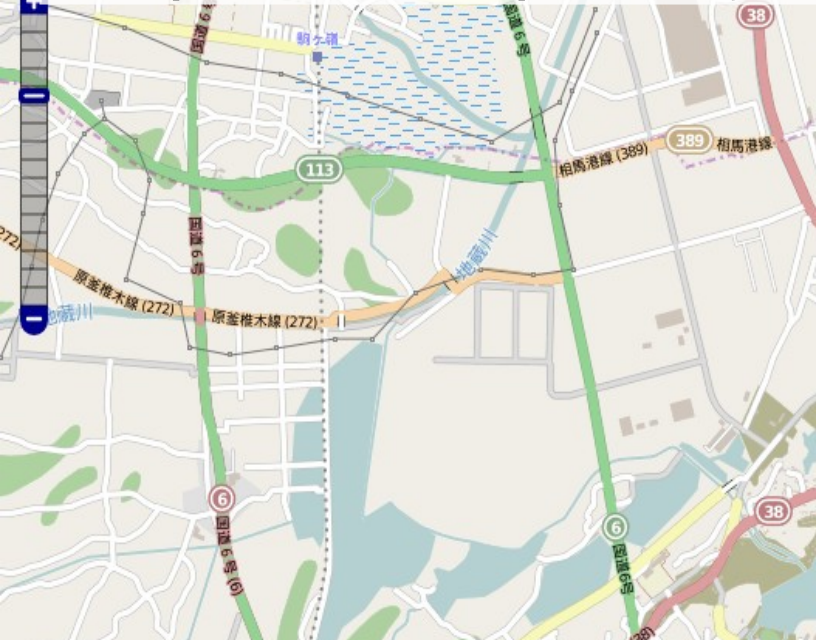
2014-01-02 - MapServer 6.4.1, 6.2.2, 6.0.4 and 5.6.9 are released

Releases for the stable branches of MapServer have been released in order to fix a potential leakage of information through an SQL injection when using TIME filtering in conjunction with postgis backends. More information can be found in the dedicated issue: [#4834](#).

Changelogs for each specific version can be found [here](#):

WMS: Web Map Service

OGCが定める地図配信の国際規格
MapServer, OpenLayers.jsを使用



施設種類	149:上屋
施設コード	BE11W
施設名	公共1号上屋

[詳細](#)



画面の地物クリック→サーバ側で地物情報を取得
→JSON形式でクライアントに返す
→クライアントがポップアップ表示



Welcome to MapServer

PostGIS(Postgresqlの地図情報拡張)やshp等を地図DBとしてWMS準拠の問い合わせにより地図を返すCGIプログラム

[More about MapServer](#)



MapServer Suite Home

[MapServer](#), [MapCache](#), [TinyOWS](#) home

proj(地図投影ライブラリ)により様々な投影座標系への変換が可能。MapServerと同時に使う。



Recent Announcements

このシステムでは地図データ座標系はWGS84(EPSG:4326, GPS座標系と同等)表示座標系はEPSG:3857(googleMap:900913と同等)

WFS: Web Feature Service

OGCが定める地図地物描画の国際規格
TinyOWS(MapServer Project)を使用

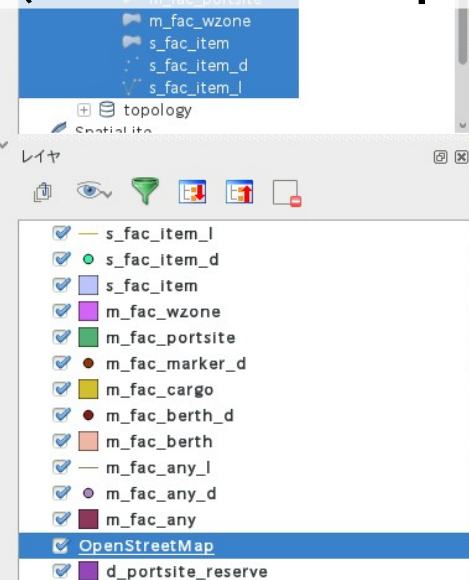


画面の編集アイコンクリックで地物を描画
→保存アイコンで地図として作成
→変更アイコンで地物の再編集が可能

QGIS：地図データ編集・解析

地図を描くだけでなく、地図の合成、データ解析も可能な統合地図アプリケーション

shp形式、PostGIS形式など直接操作可能
(小ネタ：shp形式の属性フォーマットはdbase形式)



電子地図のオープンな制作環境が整っている
今後の電子地図の活用に期待

Sunbit System



弊社の諸々も含め
紹介した考え方・実装を
汎用的なシステムに
まとめました。

自治体向け施設利用申請管理システム

My Page 船舶入

TOP 船舶入出港管理

エラー件数
2

申請・船舶入出港予実績一覧

http://amatsumi.net/

ログアウト (admin:admin)

管理

岸実績登録済、離岸実績未登録

- 新規の係留施設使用許可
- 船舶入出港予定/実績の表示
- 各データの詳細画面表示
- エラーデータの確認と修正

係留施設使用許可

- 許可対象の申請の一覧表示
- 対象データに対する許可/不許可の登録

減免申請または免除手続一覧

- 減免申請または免除手続の一覧表示
- 減免申請または免除手続の受理、許可の登録

※ フェリーの入港回数による減免についての処理は
フェリー関連処理から行って下さい

船舶PSC情報登録

- PSC情報(要注意船舶情報)の登録と一覧表示

NACCS 受付ログ

- NACCS経由の申請受付ログの一覧表示
- NACCS経由の入出港届出力

フェリー減免申請の受理、許可の登録

日毎の料金の確認と確定
カー船実績報告書の出力

詳しくは展示ブースへ

Sunbit System 有限会社サンビットシステム

- 1998年8月25日 札幌市東区に登記
- 2009年より札幌市豊平区平岸
- 今年のフレーズ：オープンをみなさんにお届け
 - ITインフラ構築、ITシステム設計支援
 - ソフトウェア受託開発,パッケージ開発
 - IT教育など
- OSC北海道には最初から協力。