

Портал самообслуживания

Версия 8.0



Эта документация предоставляется с ограничениями на использование и защищена законами об интеллектуальной собственности. За исключением случаев, прямо разрешенных в вашем лицензионном соглашении или разрешенных законом, вы не можете использовать, копировать, воспроизводить, переводить, транслировать, изменять, лицензировать, передавать, распространять, демонстрировать, выполнять, публиковать или отображать любую часть в любой форме или посредством любые значения. Обратный инжиниринг, дизассемблирование или декомпиляция этой документации, если это не требуется по закону для взаимодействия, запрещены.

Информация, содержащаяся в данном документе, может быть изменена без предварительного уведомления и не может гарантировать отсутствие ошибок. Если вы обнаружите какие-либо ошибки, сообщите нам о них в письменной форме.

Содержание

Портал самообслуживания	4
Устройство портала	4
Работа с мастером страниц на портале	4
Настройка портала и порталых пользователей	5
Ограничение доступа к веб-сервисам для пользователей портала	5
Пример использования миксина PortalMessagePublisherExtensions	8
Реализация примера	8
Изменить доступ к сервису для пользователей портала	10
Реализация примера	11
Миксин PortalMessagePublisherExtensions	11
Методы	12

Портал самообслуживания



Портал самообслуживания (Self Service Portal, SSP) является неотъемлемой функциональной частью продуктов [Service Creatio, enterprise edition](#), [Service Creatio, customer center edition](#), а также всех производных пакетов (bundle) продуктов, в которые они входят.

Портал представляет собой отдельное рабочее место (Workplace), в которое автоматически попадает пользователь портала при входе через стандартное окно авторизации приложения.

По умолчанию в поставляемом продукте порталному пользователю доступно два раздела — [*Обращения портала*] и [*База знаний*], а также [*Главная страница портала*], которая содержит общую сводную информацию и выполняет роль единого рабочего места пользователя портала.

Раздел [*Обращения портала*] служит для самостоятельной регистрации обращений пользователем, просмотра статуса своих обращений, внесения дополнительной информации по обращению, а также для получения информации по решению своего обращения. По умолчанию порталный пользователь имеет доступ только к тем обращениям, в которых он указан в качестве контакта. На странице обращения пользователь может внести дополнительную информацию, публиковать сообщения, вести переписку по своему обращению с сотрудниками службы поддержки. История переписки по обращению отображается на странице обращения на вкладке [*Обработка*].

Раздел [*База знаний*] служит для самостоятельного поиска пользователем справочной информации или решения по своему вопросу. Данный раздел по умолчанию может наполняться только сотрудниками поддержки из основного интерфейса системы.

Устройство портала

С точки зрения разработки, портал является преднастроенным отдельным рабочим местом. Это рабочее место по умолчанию недоступно обычным пользователям приложения. Пользователь системы с типом "Пользователь портала" при авторизации автоматически попадает в данное рабочее место, а именно на Главную страницу портала.

Разделы и страницы портала — это обыкновенные [разделы](#) и [страницы записи](#), и работают с теми же сущностями (`Entity`), что и страницы, доступные в основном интерфейсе системы. В частности, страница обращения на портале имеет более простой вид, и не содержит большинства полей. В конфигурации это две разные страницы записи — `CasePage` и `PortalCasePage`.

Система прав доступа на портале имеет только одно небольшое отличие. Чтобы дать порталным пользователям доступ к определенным сущностям (EntitySchema), необходимо дополнительно указать эти сущности в справочнике [*Список объектов доступных порталному пользователю*]. Лицензии портала самообслуживания ограничивают количество записей, которое может быть добавлено в данный справочник. По умолчанию это 70 записей.

Работа с мастером страниц на портале

Порталному пользователю недоступна функциональность мастеров страниц, деталей и разделов. Эта функциональность может быть вызвана пользователем с правами администратора из основного

интерфейса системы следующим образом:

1. Перейти в раздел [*Настройка рабочих мест*] дизайнера системы.
2. Выбрать рабочее место [*Портал*] и нажать [*Открыть*].
3. Выбрать требуемый раздел и нажать кнопку [*Настройка раздела*].

После этого откроется стандартный мастер разделов.

Настройка портала и порталых пользователей

Для того чтобы начать пользоваться порталом, необходимо выполнить следующие действия:

1. Убедиться, что в файле `web.config` загрузчика приложения (т. н. "внешний" `web.config`) опция `/configuration/terrasoft/auth` имеет следующий вид.

Опция `/configuration/terrasoft/auth` **файла** `web.config`

```
<terrasoft>
  <auth providerNames="InternalUserPassword,SSPUserPassword" ...>
  ...
</terrasoft>
```

Данная настройка отвечает за возможность авторизации в системе порталых пользователей.

2. Создать контакт для пользователя.
3. Создать пользователя с типом "Пользователь портала". Заполнить необходимые поля.
4. Раздать необходимые лицензии пользователю.

У порталного пользователя в профиле должен быть указан корректный часовой пояс. По умолчанию у нового пользователя он не указан. Настройка выполняется самим пользователем портала в его профиле. Отталкиваясь от этого значения, в системе будут отображаться все даты и время.

Ограничение доступа к веб-сервисам для пользователей портала

Основной сценарий использования портала — предоставить доступ большому количеству внешних по отношению к организации пользователей к ограниченному количеству данных из системы. Для решения таких задач возникает потребность управлять тем, какие пользователи (портальные или пользователи компании) имеют доступ к [веб-сервисам приложения](#).

Важно. Данная статья актуальна для версии системы 7.13.2 и новее.

Префиксы маршрутов конфигурационных веб-сервисов

Для управления доступом к веб-сервисам приложения в Creatio реализован механизм префиксов маршрута. Используя определенные `ServiceRoute` атрибуты класса сервиса, можно задать необходимый

префикс маршрута.

Префиксы маршрутов конфигурационных веб-сервисов

Доступ	Атрибут	Префикс маршрута	Пример кода
Только для пользователей Портала самообслуживания	SspServiceRoute	/ssp	<div><pre>[ServiceContract] [SspServiceRoute] public class SspOnlyService { }</pre></div> <div>Пример вызова</div> <div><pre>~/ssp/rest/SspOnlyService ~/ssp/soap/SspOnlyService</pre></div>
Только для внутренних пользователей	DefaultServiceRoute или не указывать ни один ServiceRoute атрибут	отсутствует	<div><pre>[ServiceContract] public class InternalService { }</pre></div> <div>или</div> <div><pre>[ServiceContract] [DefaultServiceRoute] public class InternalService { }</pre></div> <div>Пример вызова</div> <div><pre>~/rest/InternalService ~/soap/InternalService</pre></div>
И для внутренних пользователей, и для Портала	одновременно атрибуты SspServiceRoute и DefaultServiceRoute	/ssp или	<div><pre>[ServiceContract]</pre></div>

	<p><code>DefaultServiceRoute</code> отсутствует</p> <p>и</p> <p><code>SspServiceRoute</code></p>	<pre>[DefaultServiceRoute] [SspServiceRoute] public class CommonService { }</pre> <p>Пример вызова</p> <pre>~/rest/CommonService ~/soap/CommonService ~/ssp/rest/CommonService ~/ssp/soap/CommonService</pre>
	<p>атрибут <code>ServiceRoute</code> с указанием префикса (например, <code>"custom"</code>)</p> <p><code>ServiceRoute("custom")</code></p> <p>произвольный префикс маршрута сервиса</p>	<pre>[ServiceContract] [ServiceRoute("custom")] public class CustomPrefixS { }</pre> <p>Пример вызова</p> <pre>~/custom/rest/CustomPrefix ~/custom/soap/CustomPrefix</pre>

На заметку. Допустимо использовать одновременно несколько атрибутов `ServiceRoute`, `SspServiceRoute` и `DefaultServiceRoute`. В результате будут созданы маршруты для сервиса со всеми вариантами префиксов.

Ограничение доступа порталных пользователей к внутреннему API

Если пользователь портала (`SspUserConnection`) обратится к сервису по маршруту без префикса `"/ssp"`, сервис вернет в ответ страницу с кодом 403 (`Forbidden`).

Ограничение доступа внутренних пользователей к порталному API

Если внутренний пользователь приложения (`UserConnection`) обратится к сервису по маршруту с префиксом `"/ssp"`, сервис вернет в ответ страницу с кодом ● 403 Forbidden.

ServiceStack сервисы ядра

Методы сервисов `ServiceStack` ядра (`DataService`, `ManagerService` и другие) по умолчанию доступны только для внутренних пользователей.

Для того, чтобы какой-то из методов был доступен также и на Портале, необходимо пометить такой метод атрибутом `SspServiceAccess` - в этом случае метод будет иметь дополнительный маршрут с префиксом вида `~/DataService/ssp/...`

Если нужно, чтобы логика метода отличалась для Портала, необходимо создать новый сервис, указав для него атрибут `SspServiceAccess`, в конструктор которого передать имя оригинального метода. Например:

```
[SspServiceAccess(nameof(SelectQuery))]
public class SspSelectQuery : SelectQuery
{
}
```

создает контракт, метод которого будет зарегистрирован по пути `~/DataService/ssp/SelectQuery`.

Доступ к методам `ServiceStack` с префиксом "ssp" запрещен внутренним пользователям, а к методам `ServiceStack` без префикса "ssp" - пользователям портала.

Пример использования миксина PortalMessagePublisherExtensions

 Сложный

Пример. Использовать МИКСИН `PortalMessagePublisherExtensions`.

Реализация примера

Реализовать пример использования миксина `PortalMessagePublisherExtensions`.

PortalMessagePublisherExtensions

```
define("CaseSectionActionsDashboard", ["PortalMessagePublisherExtensions"], function() {
    return {
        mixins: {
            /**
             * @Class PortalMessagePublisherExtensions extends tabs and publishers configs.
             */
            PortalMessagePublisherExtensions: "Terrasoft.PortalMessagePublisherExtensions"
        },
        methods: {
```



```

/**
 * @inheritdoc Terrasoft.SectionActionsDashboard#getExtendedConfig
 * @overridden
 */
getExtendedConfig: function() {
    // Получение объекта конфигурации вкладок из родительского метода.
    var config = this.callParent(arguments);
    // Вызов метода миксина, добавление конфигурации портальной вкладки.
    this.mixins.PortalMessagePublisherExtensions.extendTabsConfig.call(this, config)
    // Возвращение расширенного объекта конфигурации.
    return config;
},

/**
 * @inheritdoc Terrasoft.SectionActionsDashboard#getSectionPublishers
 * @overridden
 */
getSectionPublishers: function() {
    // Получение коллекции издателей сообщений из родительского метода.
    var publishers = this.callParent(arguments);
    // Вызов метода миксина, добавление портального канала.
    this.mixins.PortalMessagePublisherExtensions.extendSectionPublishers.call(this,
    // Возвращение расширенной коллекции издателей сообщений.
    return publishers;
}
},
diff: /**SCHEMA_DIFF*/[
    {
        "operation": "insert",
        "name": "PortalMessageTab",
        "parentName": "Tabs",
        "propertyName": "tabs",
        "values": {
            "items": []
        }
    },
    {
        "operation": "insert",
        "name": "PortalMessageTabContainer",
        "parentName": "PortalMessageTab",
        "propertyName": "items",
        "values": {
            "itemType": this.Terrasoft.ViewItemType.CONTAINER,
            "classes": {
                "wrapClassName": ["portal-message-content"]
            },
            "items": []
        }
    }
]

```

```

    },
    {
      "operation": "insert",
      "name": "PortalMessageModule",
      "parentName": "PortalMessageTab",
      "propertyName": "items",
      "values": {
        "classes": {
          "wrapClassName": ["portal-message-module", "message-module"]
        },
        "itemType": this.Terrasoft.ViewItemType.MODULE,
        "moduleName": "PortalMessagePublisherModule",
        "afterrender": {
          "bindTo": "onMessageModuleRendered"
        },
        "afterrender": {
          "bindTo": "onMessageModuleRendered"
        }
      }
    }
  ]/**SCHEMA_DIFF*/
};
});

```

Изменить доступ к сервису для пользователей портала



Сложный

В приложении существует набор базовых сервисов, доступ к которым есть только у внутренних пользователей.

Для изменения доступа к базовому сервису необходимо:

1. В пользовательском пакете создать сервис с настройкой доступа для порталных пользователей.
2. Добавить для него те методы базового сервиса, которые должны быть доступны для порталных пользователей.
3. Изменить пользовательские или расширить базовые клиентские схемы, изменив вызов базового сервиса на вызов созданного сервиса (см. шаг 1).
4. Скомпилировать конфигурацию.

Пример. Изменить доступ к сервису для пользователей портала.

Реализация примера

Пример исходного кода сервиса, который расширяет доступ к базовому сервису `ActivityUtilService` представлен ниже.

PartnerActivityUtilService

```
namespace Terrasoft.Configuration
{
    using System;
    using System.IO;
    using System.Runtime.Serialization;
    using System.ServiceModel;
    using System.ServiceModel.Activation;
    using System.ServiceModel.Web;
    using Terrasoft.Configuration.FileUpload;
    using Terrasoft.Core.Factories;
    using Terrasoft.Web.Common;
    using Terrasoft.Web.Common.ServiceRouting;

    [ServiceContract]
    // Сервис доступен и для внутренних пользователей и для Портала.
    [DefaultServiceRoute]
    [SspServiceRoute]
    [AspNetCompatibilityRequirements(RequirementsMode = AspNetCompatibilityRequirementsMode.RequirementsMode.All)]
    public class PartnerActivityUtilService: BaseService {
        // Базовый сервис, доступ к которому необходимо расширить.
        private static readonly ActivityUtilService _baseService = new ActivityUtilService();

        [OperationContract]
        [WebInvoke(Method = "POST", BodyStyle = WebMessageBodyStyle.Wrapped, RequestFormat = WebMessageFormat.Json)]
        public Guid CreateActivityFileEntity(JsonActivityFile jsonActivityFile) {
            return _baseService.CreateActivityFileEntity(jsonActivityFile);
        }
        [OperationContract]
        [WebInvoke(Method = "POST", BodyStyle = WebMessageBodyStyle.Wrapped, RequestFormat = WebMessageFormat.Json)]
        public Guid CreateFileEntity(JsonEntityFile jsonEntityFile) {
            return _baseService.CreateFileEntity(jsonEntityFile);
        }
    }
}
```

Миксин PortalMessagePublisherExtensions





Миксин — это класс-примесь, предназначенный для расширения функциональности других классов. Это отдельно созданные классы с дополнительной функциональностью.

Миксин `PortalMessagePublisherExtensions` предназначен для расширения схемы `SectionActionsDashboard` и ее наследников. Он позволяет расширить конфигурацию вкладок `SectionActionsDashboard` вкладкой порталных сообщений `PortalMessageTab` и добавить соответствующий канал сообщений `Portal`. Миксин реализован в пакете `PortalMessagePublisher` и доступен в продукте `ServiceEnterprise`, а также в бандлах, в которые входит этот продукт.

Методы

`extendTabsConfig(config) : Object`

Расширяет конфигурацию вкладок `SectionActionsDashboard` вкладкой порталных сообщений `PortalMessageTab`.

Возвращает дополненный объект (`Object`) конфигурации вкладок `SectionActionsDashboard`.

Параметры

`config (Object)`

Объект конфигурации вкладок `SectionActionsDashboard`.

`extendSectionPublishers(publishers) : Array`

Добавляет порталный канал (`Portal`) в коллекцию издателей сообщений.

Возвращает дополненную коллекцию издателей сообщений (`Array`).

Параметры

`publishers (Array)`

Коллекция издателей сообщений.