ELTEX

Комплексные решения для построения сетей

Руководство по интеграции xml API

для приложения p2pTV

версия 1.2.2

Медиацентры NV

Версия документа	Дата выпуска	Содержание изменений
Версия 1.2.2	18.01.2016	Первая публикация.

Условные обозначения

Обозначение	Описание
Полужирный шрифт	Полужирным шрифтом выделены примечания и предупреждения, название глав, заголовков.
Courier New	Шрифтом Courier New в рамке с тенью указаны примеры ввода команд и результаты их выполнения.

Примечания и предупреждения

Примечания содержат важную информацию, советы или рекомендации по использованию и настройке устройства.

Предупреждения информируют пользователя о ситуациях, которые могут нанести вред устройству или человеку, привести к некорректной работе устройства или потере данных.

СОДЕРЖАНИЕ

1 АННОТАЦИЯ....4
2 СОВМЕСТИМОСТЬ....4
3 НАСТРОЙКА И КАСТОМИЗАЦИЯ ПРИЛОЖЕНИЙ....4
3.1 Настройки и кастомизация для Linux-устройств....4
3.1.1 Файл p2ptv.config....4
3.1.2 Файл p2p_desc.txt....5
3.2 Настройки и кастомизация для Android-устройств....6
4 ФОРМАТ ДАННЫХ....7
5 ВЗАИМОДЕЙСТВИЕ С СЕРВЕРОМ....7
5.1 Типы запросов....7
5.2 Параметры запросов (query params):....7
5.3 HTTP Header....8
6 ОПИСАНИЕ СТРУКТУРЫ API....8
6.1 Список категорий(tags)....8
6.2 Список фильмов для категории....9
6.3 Поиск....10
6.4 Информация о фильме....11

1 АННОТАЦИЯ

Данный документ предназначен для операторов, желающих провести интеграцию с приложением p2pTV.

На момент составления документа поддерживается две версии приложения: плагин для Linux-приставок и приложение для Android-приставок.

В данном документе описывается только взаимодействие между клиентом-афишей и сервером, а также возможности по кастомизации приложений, не описывается процесс скачивания и воспроизведения видео с хаба.

2 СОВМЕСТИМОСТЬ

Базовой версией является приложение для Linux приставок с API v2.

Приложение для Android разрабатывалось с учетом максимального сохранения совместимости с серверной частью, рассчитанной на работу с Linux-приставками. Таким образом, дополнительные правки в серверной части для поддержки работы Android-приставок не требуются.

Актуальные версии приложений на момент выпуска документа:

- Android - v2.45.
- Linux – DC.1.1.44.

3 НАСТРОЙКА И КАСТОМИЗАЦИЯ ПРИЛОЖЕНИЙ

3.1 Настройки и кастомизация для Linux-устройств

Для настройки внешнего вида и поведения плагина используются два конфигурационных файла: p2ptv.config и p2ptv_desc.txt.

3.1.1 Файл p2ptv.config

Файл **p2ptv.config** имеет следующий вид:

```
?query-params

http://path/to/tags - список категорий.

http://path/to/list/ - категория «New».

http://path/to/search/ - URL для поиска.
```

Файл p2ptv.config имеет жестко заданную структуру, перестановка элементов или добавление лишних переносов строк не допускается.

По историческим причинам для категории «New» было выделено отдельное поле http://path/to/list/. Во избежание дублирования категорий не следует добавлять категорию «New» в список категорий в поле http://path/to/tags.

3.1.2 Файл p2p_desc.txt

Файл p2p_desc.txt не имеет жестко заданной структуры.

Параметры задаются в виде связки: **ключ=значение.**

Допускается свободный порядок параметров. Каждый новый параметр должен быть на новой строке.

Приложением поддерживаются следующие параметры:

Ключ	**Тип значения**	**Описание**
desc_eng	string	Описание приложения для главного меню на английском языке.
name_eng	string	Название приложения для главного меню на английском языке.
desc_rus	string	Описание приложения на русском языке.
name_rus	string	Название приложения на русском языке.
hub_resolver	string	Адрес GEO-сервиса, предназначенного для получения адреса хаба.
enable_download	boolean	Включает возможность просмотра контента.
enable_trailers	boolean	Включает поддержку трейлеров. Автоматически отключает поддержку загрузок.
enable_filminfo	boolean	Включает поддержку информации о фильме.
main_background	string	Имя файла с изображением, которое будет использовано в качестве фона. Изображение должно лежать в одной папке с плагином.
search_preview	string	Имя файла с изображением, которое будет использовано в preview при перемещение фокуса на

		меню поиска. Изображение должно лежать в одной папке с плагином.
sd_support	boolean	Включение поддержки SD-режима.
hub	string	Адрес хаба. Игнорируется при использовании hub_resolver.
filminfo_buttons_signatures	boolean	Включение подписей для кнопок в карточке фильма.
statistics_server_url	string	Адрес сервера статистики.
enable_grouping	boolean	Включение группировки сериалов по сезонам.
provider	string	Название провайдерской библиотеки. Необходимо для работы рекламы.
api_version	string	Версия API. Поддерживаются значения 2 и 3.

Дополнительную информацию по настройке можно найти в документе «Руководство по кастомизации прошивки NV-100/102».

3.2Настройки и кастомизация для Android-устройств

Настройка приложения осуществляется стандартными средствами кастомизации – через магазин приложений и через ACS.

Ключ	**Описание**
xmlapi_tags	URL для получения списка категорий.
xmlapi_list	URL для категории «New».
xmlapi_search	URL для поиска.
xmlapi_hub	Адрес хаба с портом.
xmlapi_geo	Адрес GEO-сервиса, предназначенного для получения адреса хаба.
statistics_server	Адрес сервера статистики.
main_background_url	URL изображения для использования в качестве основного фона. Изображение будет кэшировано в основную память, при каждом запуске приложение должно проверять изменение значение и запрашивать у сервера Last-Modified для указанного изображения. В случае различий изображение будет перезагружено.
50x_message	Сообщение об ошибке, соответствующее 50x HTTP response status code.

other_error_message	Сообщение об ошибке, соответствующее остальным кодам ошибок HTTP.
color_scheme	Цветовая схема. Доступные значения: bright, dark.
api_version	Версия API. Доступны значения 2 и 3.

Дополнительную информацию по настройке можно найти в документе «Руководство по установке и работе с магазином приложений».

4 ФОРМАТ ДАННЫХ

Взаимодействие между клиентом и сервером ведется по протоколу HTTP.

В качестве ответов на запросы к API клиент должен получать файлы в XML-формате, структура которых описана ниже. Допускается наличие произвольных полей в полученных файлах.

5 ВЗАИМОДЕЙСТВИЕ С СЕРВЕРОМ

5.1 Типы запросов

В XML API реализовано три типа запросов:

- запрос списка категорий («Мелодрама», «Комедия», «Ужасы» и т. д.).
- запрос списка фильмов.
- поисковый запрос.

Для Linux-устройств URL для запросов указываются в конфигурационном файле p2ptv.config.

Для Android-устройств URL задаются через механизмы кастомизации магазина приложений и ACS.

5.2 Параметры запросов (query params):

Список параметров:

- *api-version* – версия XML API, в настоящее время поддерживаются версии 2 и 3.
- *device* – модель устройства (например, fs_nv101_111124).

Для моделей NV-10x и NV-30x параметры api-version и device можно задать в конфигурационном файле p2ptv.config.

- *page-size* – количество элементов на странице.
- *p* – номер запрашиваемой страницы начиная с 1.

- *tag* – имя категории.
- *q* – строка поиска.

5.3 HTTP Header

При запросе xml-элементов в Header GET запросов добавляется User-Agent вида:

device/firware_version (s/n,MAC)

Пример:

```
Eltex NV-101/fs_nv102_141119_0.0.87 (SB0A014926, A8:F9:4B:22:BE:4D)
```

6 ОПИСАНИЕ СТРУКТУРЫ API

Для описания структуры используется язык XML Schema.

6.1 Список категорий(tags)

Список категорий предназначен для формирования главного меню приложения.

Поля *«Search»* и «*New»* будут добавлены автоматически.

Поле «*URL»* должно указывать на список фильмов для данной категории.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
 <xs:element name="Tags">
  <xs:complexType>
  <xs:sequence>
            <xs:element ref="Items"/>
  </xs:sequence>
  </xs:complexType>
 </xs:element>
 <xs:element name="Items">
  <xs:complexType>
  <xs:sequence>
            <xs:element ref="Item"/>
  </xs:sequence>
  </xs:complexType>
 </xs:element>
 <xs:element name="Item">
  <xs:complexType>
  <xs:sequence>
            <xs:element name="Name" type="xs:string"/>
            <xs:element name="Title" type="xs:string"/>
            <xs:element name="URL" type="xs:string"/>
  </xs:sequence>
  </xs:complexType>
 </xs:element>
</xs:schema>
```

6.2 Список фильмов для категории

Список параметров:

- *Page* – номер текущей страницы.
- *PagesCount* – общее количество страниц.
- *Poster* – URL-миниатюра постера фильма для API v2.
- *Preview* – URL-миниатюра постера фильма для API v3.
- *Year* – год выпуска фильма, отрисовывается на иконке фильма для Android-приставок.
- *TItle* – название фильма.
- *Url* – ссылка на информацию о фильме. Используется для отображения карточки фильма.

Размеры изображений:

- Linux - 200px:267px – изображения не масштабируются.
- Android - 200dp:255dp – изображения масштабируются.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
 <xs:element name="Films">
  <xs:complexType>
  <xs:sequence>
              <xs:element name="Page" type="xs:string"/>
              <xs:element name="PagesCount" type="xs:string"/>
              <xs:element ref="Items"/>
  </xs:sequence>
  </xs:complexType>
 </xs:element>
 <xs:element name="Items">
  <xs:complexType>
  <xs:sequence>
              <xs:element ref="Item"/>
  </xs:sequence>
  </xs:complexType>
 </xs:element>
 <xs:element name="Item">
  <xs:complexType>
  <xs:sequence>
              <xs:element name="URL" type="xs:string"/>
              <xs:element name="Title" type="xs:string"/>
              <xs:element name="Year" type="xs:string"/>
              <xs:element name="Preview" type="xs:string"/>
              <xs:element name="Poster" type="xs:string"/>
  </xs:sequence>
  </xs:complexType>
 </xs:element>
</xs:schema>
```

6.3Поиск

Для осуществления поиска используется URL вида:

{xmlapi_search}?{query_params}&page_size={n}&q={search_word}

Где:

- *xmlapi_search* – URL поиска, заданный через стандартные механизмы настройки приложения.
- *query_params* – параметры запроса. Задаются в конфигурационном файле для Linux-приставок. Для Android-приставок для API v2 используется строка:
  ?device=eltex-nv101-mkv&api-version=2
  Для API v3 используется реальные данные
- *n* – требуемое количество элементов (Item) в ответе сервера. Допускается произвольное количество на усмотрение сервера.
- *search_word* – слово для поиска.

Список параметров:

- *Page* – номер текущей страницы.
- *PagesCount* – общее количество страниц.
- *Poster* – URL-миниатюра постера фильма для API v2.
- *Preview* – URL-миниатюра постера фильма для API v3.
- *Year* – год выпуска фильма, отрисовывается на иконке фильма для Android-приставок.
- *TItle* – название фильма.
- *Url* – ссылка на информацию о фильме. Используется для отображения карточки фильма.
- *Tags* – список тегов для фильма. Опционально.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
 <xs:element name="Films">
  <xs:complexType>
  <xs:sequence>
              <xs:element name="Page" type="xs:string"/>
              <xs:element name="PagesCount" type="xs:string"/>
              <xs:element ref="Items"/>
              <xs:element ref="Tags"/>
  </xs:sequence>
  </xs:complexType>
 </xs:element>
 <xs:element name="Items">
  <xs:complexType>
  <xs:sequence>
```

```
            <xs:element ref="Item"/>
  </xs:sequence>
  </xs:complexType>
 </xs:element>
 <xs:element name="Item">
  <xs:complexType>
  <xs:sequence>
            <xs:element name="URL" type="xs:string"/>
            <xs:element name="Title" type="xs:string"/>
            <xs:element name="Year" type="xs:string"/>
            <xs:element name="Preview" type="xs:string"/>
            <xs:element name="Poster" type="xs:string"/>
  </xs:sequence>
  </xs:complexType>
 </xs:element>
 <xs:element name="Tags">
  <xs:complexType>
  <xs:sequence>
            <xs:element ref="Tag"/>
  </xs:sequence>
  </xs:complexType>
 </xs:element>
 <xs:element name="Tag">
  <xs:complexType>
  <xs:sequence>
            <xs:element name="Title" type="xs:string"/>
            <xs:element name="URL" type="xs:string"/>
  </xs:sequence>
  </xs:complexType>
 </xs:element>
</xs:schema>
```

6.4Информация о фильме

Запрос информации о фильме по ссылкам, полученным следующих источников:

- поле Url из списка фильмов для категории.
- поле Url из списка актеров/режиссеров.
- поле Url из списка, полученного при поисковом запросе.
- поле Url из списка SeeAlso.

Список параметров:

- *Poster* – URL-постер фильма. Размеры виджета для отображения постера - 270px:400px на Linux-приставках, изображение не масштабируется. На Android-приставках - 300dp:380dp, изображение масштабируется.
- *Year* – год выпуска фильма. Отрисовывается на иконке фильма для Android-приставок.
- *TItle* – название фильма.
- Tags – список тегов для фильма. Опционально.
- *Id* – используется для отправки статистики.

- *Directors/Actors* – используется для отображения списка актеров/режиссеров. Открывается по нажатию на кнопку *«Инфо»* на карточке фильма. Структура данных соответствует списку фильмов для категории.

- *Items* – список серий. Если элементов Item несколько - фильм считается сериалом и по нажатию на кнопку *«Воспроизвести»* откроется список выбора серии. Если элемент один - запустится воспроизведение.

- *Item* – представляет информацию об одной серии фильма. Поле *«Id»* необходимо для отправки статистики. Поле *«Group»* используется для группировки фильмов по категориям.

  На Android-приставках группировка включена по умолчанию. На Linux-приставках ее необходимо включить в конфигурации.

- *Trailers* – используется для отображения трейлеров. На Android-приставках поддержка трейлеров включена по умолчанию. На Linux-приставках ее необходимо включить в конфигурации. **При включении данного параметра автоматически отключается поддержка загрузок.**

- *Ratings* – используется для отображения рейтинга. Поле *«Type»* используется только на Linux-приставках. Если его значение равно **cn** - для отрисовки рейтинга будут использованы символы-звездочки. На Android-приставках символы-звездочки используются для отображения любых элементов рейтинга.

- SeeAlso – используется для отображения списка похожих фильмов. Структура данных соответствует списку фильмов для категории. Размеры постеров: Linux - 120px:160px, изображение масштабируется; Android - 140dp:150dp, изображение масштабируется.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
 <xs:element name="Films">
  <xs:complexType>
  <xs:sequence>
            <xs:element ref="Film"/>
  </xs:sequence>
  </xs:complexType>
 </xs:element>
 <xs:element name="Film">
  <xs:complexType>
  <xs:sequence>
            <xs:element name="Id" type="xs:string"/>
            <xs:element name="Title" type="xs:string"/>
            <xs:element name="Desc" type="xs:string"/>
            <xs:element name="Year" type="xs:string"/>
            <xs:element name="Preview" type="xs:string"/>
            <xs:element name="Poster" type="xs:string"/>
            <xs:element ref="Tags"/>
            <xs:element ref="Directors"/>
            <xs:element ref="Actors"/>
            <xs:element ref="Items"/>
            <xs:element ref="Ratings"/>
  </xs:sequence>
  </xs:complexType>
 </xs:element>
 <xs:element name="Tags">
```

```
 <xs:complexType>
 <xs:sequence>
            <xs:element ref="Tag"/>
 </xs:sequence>
 </xs:complexType>
</xs:element>
<xs:element name="Tag">
 <xs:complexType>
 <xs:sequence>
            <xs:element name="Id" type="xs:string"/>
            <xs:element name="Name" type="xs:string"/>
            <xs:element name="Title" type="xs:string"/>
            <xs:element name="URL" type="xs:string"/>
 </xs:sequence>
 </xs:complexType>
</xs:element>
<xs:element name="Directors">
 <xs:complexType>
 <xs:sequence>
            <xs:element ref="Director"/>
 </xs:sequence>
 </xs:complexType>
</xs:element>
<xs:element name="Director">
 <xs:complexType>
 <xs:sequence>
            <xs:element name="Id" type="xs:string"/>
            <xs:element name="Title" type="xs:string"/>
            <xs:element name="URL" type="xs:string"/>
 </xs:sequence>
 </xs:complexType>
</xs:element>
<xs:element name="Actors">
 <xs:complexType>
 <xs:sequence>
            <xs:element ref="Actor"/>
 </xs:sequence>
 </xs:complexType>
</xs:element>
<xs:element name="Actor">
 <xs:complexType>
 <xs:sequence>
            <xs:element name="Id" type="xs:string"/>
            <xs:element name="Title" type="xs:string"/>
            <xs:element name="URL" type="xs:string"/>
 </xs:sequence>
 </xs:complexType>
</xs:element>
<xs:element name="Items">
 <xs:complexType>
 <xs:sequence>
            <xs:element ref="Item" maxOccurs="unbounded"/>
 </xs:sequence>
 </xs:complexType>
</xs:element>
<xs:element name="Item">
 <xs:complexType>
 <xs:sequence>
            <xs:element name="Id" type="xs:string"/>
            <xs:element name="URL" type="xs:string"/>
            <xs:element name="Shown" type="xs:string"/>
            <xs:element name="Title" type="xs:string"/>
            <xs:element name="Magnet" type="xs:string"/>
            <xs:element name="Group" type="xs:string"/>
            <xs:element name="Year" type="xs:string"/>
```

```
            <xs:element name="Preview" type="xs:string"/>
            <xs:element name="Poster" type="xs:string"/>
  </xs:sequence>
  </xs:complexType>
 </xs:element>
 <xs:element name="Ratings">
  <xs:complexType>
  <xs:sequence>
            <xs:element ref="SeeAlso"/>
            <xs:element ref="Trailers"/>
            <xs:element ref="Rating"/>
  </xs:sequence>
  </xs:complexType>
 </xs:element>
 <xs:element name="SeeAlso">
  <xs:complexType>
  <xs:sequence>
            <xs:element ref="Item"/>
  </xs:sequence>
  </xs:complexType>
 </xs:element>
 <xs:element name="Trailers">
  <xs:complexType>
  <xs:sequence>
            <xs:element ref="Item"/>
  </xs:sequence>
  </xs:complexType>
 </xs:element>
 <xs:element name="Rating">
  <xs:complexType>
  <xs:sequence>
            <xs:element name="Type" type="xs:string"/>
            <xs:element name="Title" type="xs:string"/>
            <xs:element name="Count" type="xs:string"/>
            <xs:element name="Max" type="xs:string"/>
            <xs:element name="Rank" type="xs:string"/>
  </xs:sequence>
  </xs:complexType>
 </xs:element>
</xs:schema>
```