



## D6.4 Websites for releases

<i>File name</i>	PuppyIR_D6.4_Websites_for_releases
<i>Author(s)</i>	Irene Schmidt (ATOS), José Miguel Garrido (ATOS), Jeldrik Smuch (ATOS)
<i>Work package/task</i>	WP6/T6.5
<i>Document status</i>	Final
<i>Contractual delivery date</i>	M30
<i>Confidentiality</i>	Public
<i>Keywords</i>	Dissemination, product, website, development, open source releases
<i>Abstract</i>	This document describes the web spaces used for the promotion of the open source releases of the PuppyIR framework and the corresponding documentation.

## Table of Contents

Executive Summary.....	3
1 Introduction .....	4
2 The PuppyIR webpages.....	5
2.1 General web site .....	5
2.2 Repository for code .....	5
3 Documentation for system designers and developers.....	7

## Executive Summary

This document describes the web spaces used for the promotion of the open source releases of the PuppylR framework and the corresponding documentation..

The structure of the document is as follows: first, we explain the rationale for a web-based presence for the code and documentation developed according to the PuppylR framework, then we describe the various aspects of the website intended to improve the awareness of PuppylR in the open source community, and finally we describe the structure and contents of the website.

# 1 Introduction

One of the main objectives of PuppylR is to provide a suite of components that can be used by external designers and developers to construct customized information retrieval systems. The dissemination and exploitation strategy is targeted towards user communities and the open source community. As also described in D6.2, D6.3 and other documents about the exploitation strategy, a key element of PuppylR success is to make an impact in the open source community.

From the beginning, PuppylR was conceived as an open source development initiative. The source code created as part of WP4 is therefore available for download from sourceforge, a widely known place to store open source software, using a specialized tool like Subversion in the case of PuppylR repository. But in addition to giving access, web-based presence is also needed for other reasons such as awareness raising and documentation.

To attract the attention of new stakeholders and developers and to raise their interest in the uptake and use of the PuppylR code easily accessible showcases are needed to exhibit the findings and achievements.

In theory it is possible to understand a program or software library based on the source code only, but in practice this is usually really hard. Lack of public available documentation is one of the factors that can discourage the open source developers. In order to make PuppylR a persistent success, we need to provide introductory documents, tutorials, API references and helpful material in general.

All these reasons make it desirable that the general PuppylR project webpage is linked to web space for promoting the use of the source code and for demonstrating the experimental assets from PuppylR.

According to the DoW (created in 2008), PuppylR could produce a DVD and/or website. In 2012 producing a DVD seems no longer to have any added value over providing web-based access to demonstrators and code.

## 2 The PuppyIR webpages

### 2.1 General web site

The webpage for releases and resources for developers is part of the general website of PuppyIR (<http://www.puppyIR.eu>). This general website displays information about the project, and has a news and events section and a list of related publications for everybody who wants to study the findings of PuppyIR in depth. The assets for open source community are centred on a main index page ([www.puppyIR.eu/framework](http://www.puppyIR.eu/framework)) and includes all kinds of documentation, which is further described in Section 3.



Figure 1: Screen dump of PuppyIR general website

### 2.2 Repository for code

A second resource used by PuppyIR is the project webpage in SourceForge, is a web-based source code repository. It acts as a centralized location for software developers to control and manage open source software development. The PuppyIR code can be accessed through <http://sourceforge.net/projects/puppyir/>.

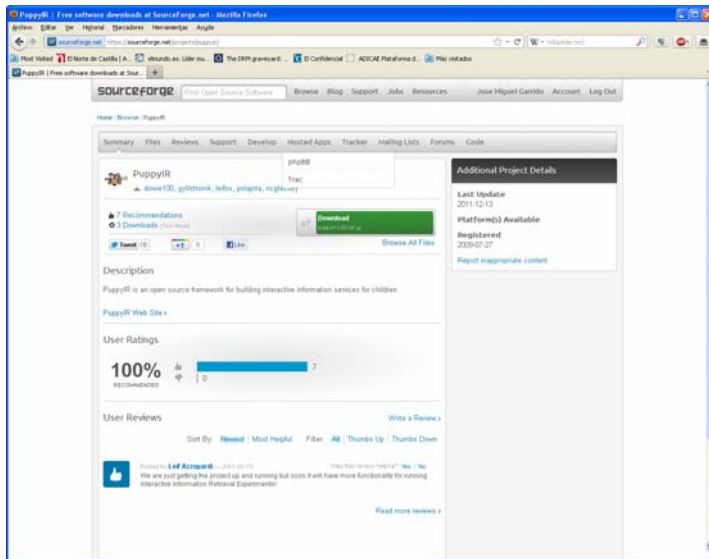


Figure 2: Screen dump of PuppyIR SourceForge opening page

### 3 Documentation for system designers and developers

As said the assets for users of the open source software developed in the project are centred on a main index: <http://www.puppyIR.eu/framework>

It is an entry point for developers and all other parties interested in using the project results.

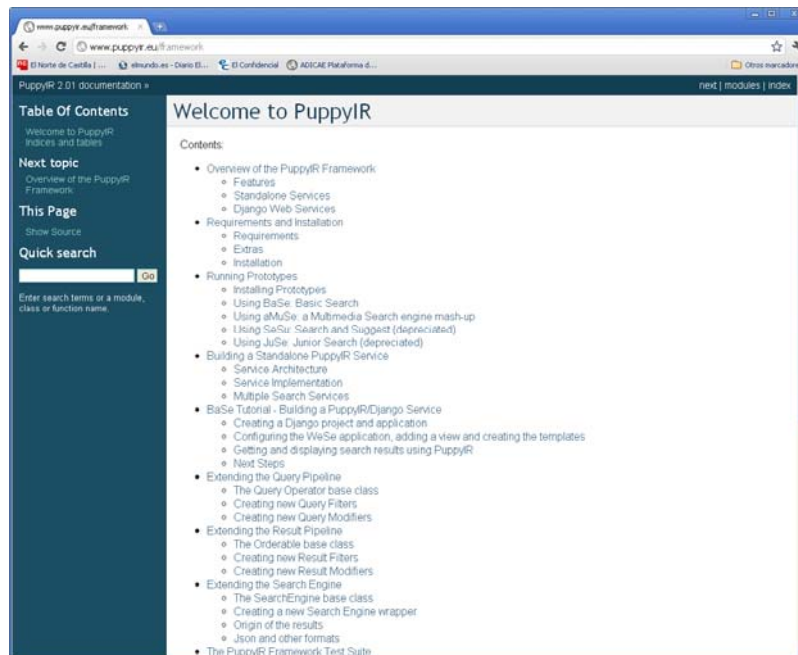


Figure 3: Screen dump of Welcome page for PuppyIR documentation page

This page is constructed using Sphinx (<http://sphinx.pocoo.org/>). This tool is an automated documentation generator specifically indented for Python-based software, and it is a very convenient way of writing the contents. (Sphinx is almost a standard for projects using Python.)

By using Sphinx, we can extract the comments directly from the source code, the same way as Javadoc does it for Java or Doxygen for C. Sphinx also provides a search engine for keywords inside the documentation.

The documentation is split up into several sections: the 'About' section, which details the design of the framework and how the various components relate to each other; the 'Tutorials' section, which provides practical examples of using the framework for a variety of audiences; the 'Extending' section details how the framework can be extended to add new filters, modifiers and search engine wrappers; the 'Appendices' section which details various supplementary materials that further expand on the other sections including an FAQ (frequently asked questions); and, finally, the API reference which details the components in framework (including details about their parameters etc).

Below we highlight a few elements:

**Overview of PuppyIR Framework:** a general introduction to the structure of PuppyIR and the framework for newcomers to PuppyIR

**Requirements and Installation:** a very important section for developers and people interested in the framework and demo applications. The installation of the framework is not difficult because it is similar to any standard python packet. But it integrates several open source libraries which are prerequisites.

**Running prototypes:** Instructions for installing and running some of the prototypes developed as demonstrators of the PuppylR framework

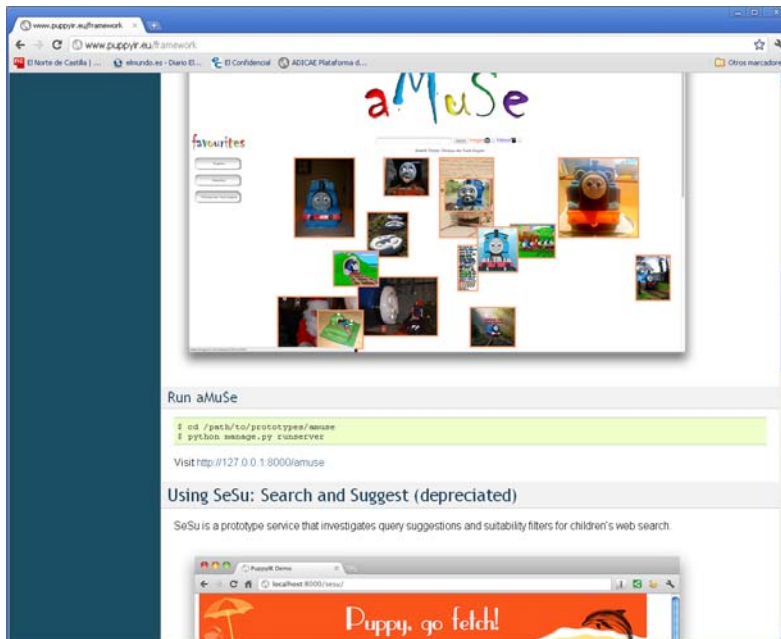


Figure 4: Screen dump of information on PuppylR prototypes

**Building a standalone PuppylR service:** here we have a tutorial about the library basic architecture, and a step-by-step tutorial about creating a simple search application without graphical interface.

**BaSe Tutorial – Building a Puppy/Django Service:** After the tutorial about constructing a standalone service using the PuppylR framework, we learn about creating a search application using Django as a framework for web applications.

**Extending the Query Pipeline, the Result Pipeline and Search Engine:** The PuppylR Framework is designed using a modular architecture and it can be extended. The PuppylR Framework includes several filters both for the queries and results, and search engine wrappers for the most common search engines. But a developer can easily create new filters and wrappers. This section contains a description of the programming interface for filters and wrappers and some examples of each kind.

**The PuppylR test suite:** The PuppylR Framework includes a built-in test suite for creating automated unit tests. In this tutorial, it is explained how to use it and how to create new tests.

**IfSe tutorial: Information Forage Search Application:** This a step-by-step tutorial about another graphical search application using Django. This application is a more advanced version of BaSe and some sophisticated aspects of the framework are explained.



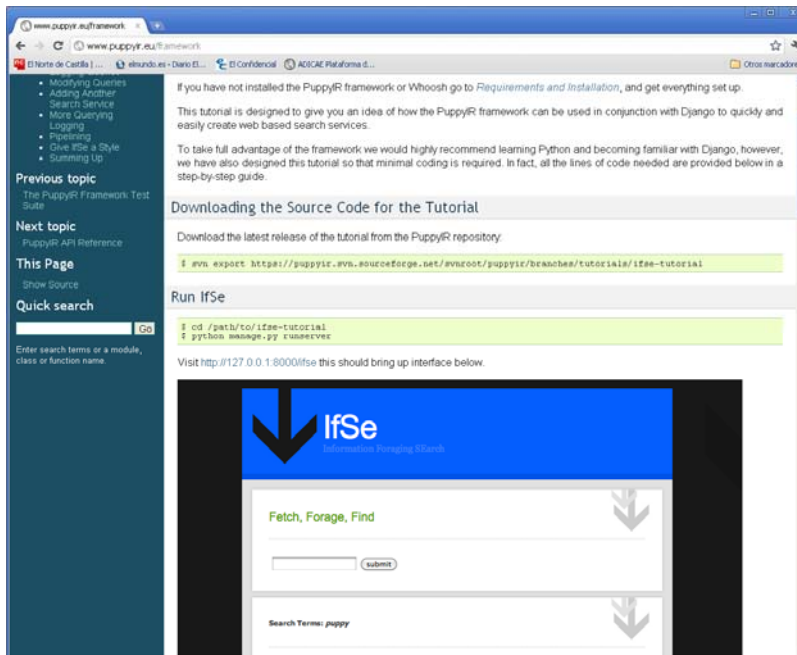


Figure 5: Screen dump of PuppyIR Tutorial pages

**PuppyIR API reference:** A complete reference of the programming interface of the Framework. This is essential for developers using the PuppyIR Framework.

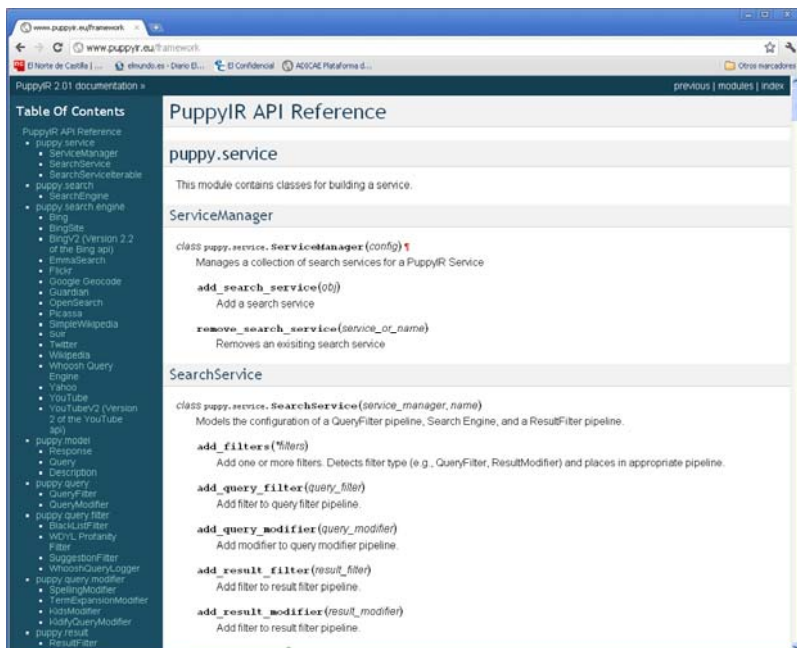


Figure 6: Screen dump of PuppyIR API reference