SETS®
SOLUTIONS

TECHNICAL SPECIFICATION DOCUMENT

RECRUITMENT PORTAL

PREPARED BY THE ENTERPRISE SOFTWARE DEVELOPMENT DEPARTMENT

Table of Contents

Document Purpose

The document hereby is a technical description of People365 recruitment portal solution. After reading it you will be able to understand the architecture, deployment, and API of the recruitment portal approach.

This document is addressed to the Information technology team specifically for system administrators and security teams.

Introduction

The recruitment portal is the new face of the careers and application form filling process of people365. This new interface is powered by People365, and is developed to gather applicant data directly from the client's website.

System Architecture

Architecture diagram
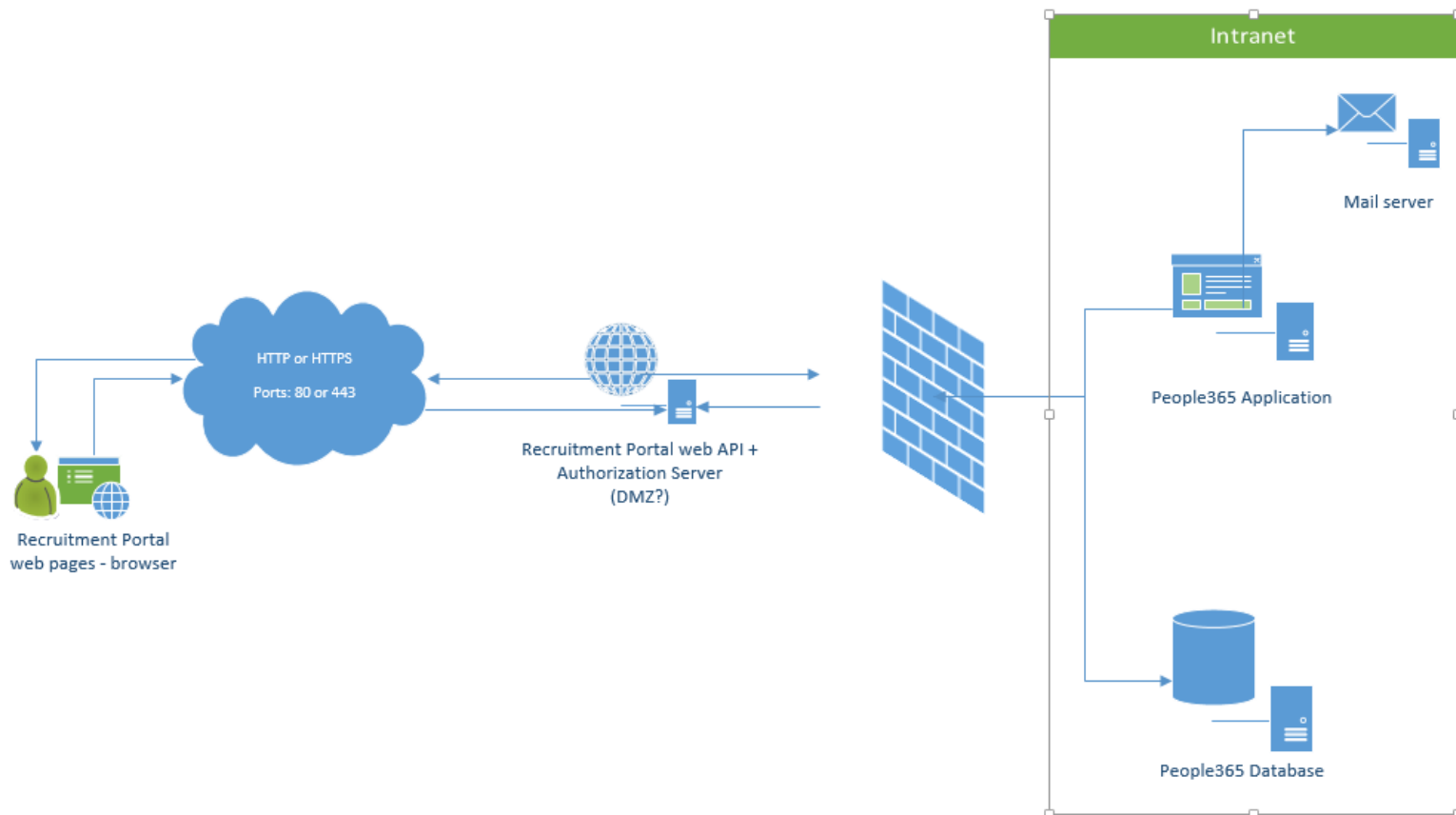
The recruitment portal web pages are recommended to be implemented using bootstrap, and should be compatible with angular JS 1.0.

The recruitment Portal web API and authorization server should be accessible from the internet. Also the Web API should have access to People365 application, and database server. This installation can live in a DMZ and behind a firewall.

Application Programming Interface (API)

Server-side web API is built using of the ASP.Net framework. Its purpose is for building HTTP services that act as an interface for the outside world to interact with the business logic. This internal business logic and the data associated with it are kept secure - information outside the provided services is not to be exposed. These HTTP services can be consumed by a broad range of clients including browsers, mobiles, iPhone and tablets.

For the recruitment portal we have 32 web services:

ID	Type	URI	Description
1	Post	/api/ApplicationForms	Save Applicant Application form
2	Get	/api/ApplicationForms/Get?personId={applicantID}	Load applicant data
3	Get	/api/BloodType	Get list of blood types
4	Get	/api/Countries	Get list of countries
5	Get	/api/Currencies	Get list of currencies
6	Get	/api/EducationalInstitutions	Get list of Educational institutions
7	Get	/api/EducationalInstitutionTypes	Get list of educational institution types
8	Get	/api/EducationDegrees	Get list of education degrees
9	Get	/api/Jobs	Get list of Jobs desired
10	Get	/api/Language	Get list of languages
11	Get	/api/MaritalStatuses	Get list of marital statuses
12	Get	/api/Nationalities	Get list of nationalities
13	Get	/api/OrganizationUnits	Get list of published OUs
14	Get	/api/RatingScale	Get list of rating scale for languages
15	Get	/api/RecruitmentSources	Get list of recruitment sources
16	Get	/api/RecruitmentSourceTypes/?isApplicant={isApplicant}	Get list of recruitment source types
17	Get	/api/Schedules	Get list of preferred schedules
18	Get	/api/UploadFile/DeleteFile?path={fileName}	Delete applicant CV

19	Get	/api/UploadFile/getFileSettings	Check settings for upload CV
20	Post	/api/UploadFile/UploadFiles	Upload applicant CV
21	Get	/api/UserAccounts/?userName={userName}	Validate user before accessing application form
22	Post	/api/Authorize	Authenticate user and provide access token
23	Get	/api/Jobs/DropDown	Get list of jobs for filtering
24	Get	/api/Jobs/DropDownCategories	Get list of jobs categories for filtering
25	Get	/api/Jobs/FiltredViewCareerGridList?categoryId={categoryIDtoFilter}&jobID={JobIDtoFilter}	Filter vacancies based on Job and job category
26	Get	/api/Jobs/SendJob?from={fromName} &to={emailreceiver} &message={messageTosend} &jobVacId={VacancyIDreferred}	Send email to friend to refer a vacancy (email will be sent)
27	Get	/api/Jobs/VacancyDetails?VacancyID={vacancyID}	Get job vacancy details
28	Get	/api/Jobs/ViewCareerGridList	Get list of job published vacancies
29	Post	/api/UserAccounts	Create a new user account and send email for activation
30	Post	/api/UserAccounts/Activate	Redirection from registration email to activate created user
31	Post	/api/UserAccounts/changePassword	Redirection from link of forget password email to update user password
32	Post	/api/UserAccounts/ForgetPassword	Send user an email to recover password

Functionality breakdown

Authorization

This system implements a token based authentication using Owin OAuth 2.0. The OAuth 2.0 framework enables our recruitment app to obtain limited access to the recruitment web API using an access token (which is a string denoting a specific scope, lifetime, and other access attributes).

We use People365 Authorization server for granting access to the People365 Recruitment Portal Web API. The http services consumed for this are ID 21, and 22 above.

Registration

When a user registers to the recruitment portal in order to apply for a vacancy, People365 will send him an email with a verification and activation link. If the user did not activate his user, he will not be able login and apply for a vacancy. Upon activation an applicant user is created in People365. This user can now be used to fill and save application forms. The http service consumed for this is 29, and 30 above.

Forget password

In case a user forgets his password, he can use the forget password functionality to recover it. When he requests to the change his password, People365 will send him an email with a recovery link. Upon recovery the system will allow him to enter a new password. The http service consumed for this is ID 31, and 32 above.

Job Posting

The user can view all vacancies published on the careers page of the recruitment portal. The http service used is ID 28 above.

The user can filter this data based on jobs or categories. The http service consumed for this is ID 23, 24, and 25 above.

The applicant also can view the details of a job vacancy. This details loaded are reference number, job title, organization unit name, work location, number Of Openings, date needed, and details. The http service consumed for this is ID 27 above.

The applicant can also refer a job vacancy to a friend by email. The email will be sent by People365. The http service consumed for this is ID 26 above.

Fill Application form

Upon successfully logging in, the applicant can now fill all the sections of the application form provided. And upload his CV. The CV document will be converted to bytes and saved along with his profile. The http service consumed for this is ID 1 – 20 above.