**Exposable Service Methods:**

Service Method needs to be call from UI for populating and saving the data.

**My Dossier**:

1. My Dossier section will display the View mode for Resource details. Below service method need to be call for populating the details. Service methods need to added under ‘namespace **ZeroChaos.Service.Services**’ and assembly **ZeroChaos.Service**
   1. ResourceDossierResponse GetResourceDossierResponseByResourceID (int resourceID);
      1. Input parameter: ResourceID (int)
      2. Output: ResourceDossierResponse (entity)
      3. Method Name: GetResourceDossierResponseByResourceID
      4. Service: ResourceService
      5. Description: This method exists in ResourceService.cs. Method will be used to populate all section in My Dossier view. Display below sections by using ResourceDossierResponse object.
         1. Career Goals / Objective: From Summary property display CareerGoal, AspiringQuote and ProfileImageName.
         2. Contact Information: Use Resource property. Address is a combination of city, state and pincode.
         3. Highest Education: This section will use ResourceEducation property, which pulls highest education.
         4. Summary: use property Summary. This contains active projects, past projects and active submissions count, list of skills and collection of work experiences.
         5. Languages: use property ResourceLanguageProficiency, to set data use ResourceLanguageProficiencyService.GetResourceLanguageProficiencyByResourceID.
         6. Personal information: Use Resource property to populate all data. To load data use GetResourceDetailByResourceID() from ResourceService.cs.
   2. ResourcePersonalInformationResponse GetResourcePersonalInformationResponseByResourceID(int resourceID);
      1. Input parameter: ResourceID (int)
      2. Output: ResourcePersonalInformationResponse(entity)
      3. Method Name: GetResourcePersonalInformationResponseByResourceID,
      4. Service: ResourceService.cs
      5. Description: Set Resource property using GetResourceDetailByResourceID method from ResourceService.cs under namespace ZeroChaos.Service.Services.
   3. ResourceJobPreferenceResponse GetResourceJobPreferenceResponseByResourceID(int ResourceID);
      1. Input parameter: ResourceID (int)
      2. Output: ResourceJobPreferenceResponse(entity)
      3. Service: ResourceJobPreferenceService.cs
      4. Method Name: GetResourceJobPreferenceResponseByResourceID
      5. Description: below are properties:
         1. ResourceJobPreference: It holds job preference information which is loaded using GetResourceJobPreferenceByResourceID() method..
         2. ResourceLegallyCountryToWork: Legally countries to work.This contains list of selected countries, need to display them as per mock. Load data using method GetResourceJobPreferenceCountryByJobPreferenceIDResourcePreferenceForCountryID() from ResourceJobPreferenceCountryService.cs.
         3. ResourcePreferredCountryToWork: Load preferred countries to work.
            1. Service: ResourceJobPreferenceCountryService.cs
            2. Method :GetResourceJobPreferenceCountryByJobPreferenceIDResourcePreferenceForCountryID().
         4. ResourceJobClassifications: Job classification selected will be loaded into it.
            1. Service: ResourceJobClassificationService
            2. Method: GetResourceJobClassificationByResourceId(resourceid)
         5. ResourceIndustries: Selected industries will be loaded into it.
            1. Service: ObjectIndustryMappingService
            2. Method: GetObjectIndustryMappingByObjectIDPKID().
   4. ResourceProjectReviewResponse GetResourceProjectReviewResponseByResourceID(int ResourceID);
      1. Input parameter: ResourceID (int)
      2. Output: ResourceProjectReviewResponse (entity)
      3. Method Name: GetResourceProjectReviewResponseByResourceID
      4. Description: Load all survey responses.// TO DO LIST
   5. ResourceWorkExperienceResponse GetResourceWorkExperiencesResponseByResourceID(int resourceID)
      1. Input parameter: ResourceID (int)
      2. Output: ResourceWorkExperienceResponse (entity)
      3. Method Name: GetResourceWorkExperiencesResponseByResourceID
      4. Description: Loads work experience and project experience.
         1. ResourceWorkExperiences: Loads all work experiences.
            1. Service: ResourceWorkExperienceService
            2. Method: GetResourceWorkExperienceByResourceID
         2. for each work experience we need to get ObjectIndustryMappings and ObjectJobCategoryMappings
            1. Service to load industry mappings: ObjectIndustryMappingService
            2. Method: GetObjectIndustryMappingByObjectIDPKID
            3. Service to load job categories: ObjectJobCategoryMappingService
            4. Method: GetObjectJobCategoryMappingByObjectIDPKID
         3. ProjectExperiences: Load project experiences.
   6. ResourceJobSkillCertificationResponse GetResourceJobSkillCertificationResponseByResourceID(int resourceID)
      1. Input parameter: ResourceID (int)
      2. Output: ResourceJobSkillCertificationResponse (entity)
      3. Service: ResourceJobSkillService
      4. Method Name: GetResourceJobSkillCertificationResponseByResourceID
      5. Description: Below properties used to display skills and certifications.
         1. ResourceJobSkills: load data from below service method:
            1. Service: ResourceJobSkillService
            2. Method: GetResourceJobSkillByResourceID()
         2. ResourceCertifications: Below are the service methods.
            1. Service: ResourceCertificationService
            2. Method: GetResourceCertificationByResourceID()
   7. ResourceResumeAttachmentResponse GetResourceResumeAttachmentResponseByResourceID(int resourceID)
      1. Input parameter: ResourceID (int)
      2. Output: ResourceResumeAttachmentResponse (entity)
      3. Service: ResourceService.cs
      4. Method Name: GetResourceResumeAttachmentResponseByResourceID
      5. Description:
         1. ResourceResume: Load data using below service method.
            1. Service:ResourceResumeService
            2. Method: GetResourceResumeByResourceID
         2. ResourceAttachment: get list of attachments using below service methods.
            1. Serive: AttachmentMappingService
            2. Method: GetAttachmentMappingsByPKIDObjIDEventAttachmentID
         3. ResourceWebReferences: load list of all webreferences.
            1. Service: ResourceWebReferencesService
            2. Method: GetResourceWebReferencesByResourceID
   8. ResourceEducationLanguageResponse GetResourceEducationLanguageResponseByResourceID(int resourceID)
      1. Input parameter: ResourceID (int)
      2. Output: ResourceEducationLanguageResponse (entity)
      3. Service: ResourceEducationService
      4. Method Name: GetResourceEducationLanguageResponseByResourceID
      5. Description:
         1. ResourceEducation: Load data using below service methods.
            1. Service: ResourceEducationService
            2. Method: GetResourceEducationByResourceID
         2. ResourceLanguageProficiency: load all language proficiencies using below service methods.
            1. Service: ResourceLanguageProficiencyService
            2. Method: GetResourceLanguageProficiencyByResourceID

**Edit Dossier:**

Edit Dossier section will Insert/Update the resource record. Below service methods need to be call for Inserting /Updating the resource details.

1. int AddUpdateResourcePersonalInformationByResourceID (ResourceRequest resourceRequest);
   1. Input parameter: ResourceRequest(entity)
   2. Output: int
   3. Method Name: AddUpdateResourcePersonalInformationByResourceID
   4. Description: Load resource details using method ‘GetResourceDetailByResourceID(resourceID)’. In update mode call resource update service to update top section and personal info tab section details.
      1. Service: ResourceService
      2. Method: UpdateResource
2. Job preference tab:
   1. Get: Load job preference details using below service methods:
      1. Service:ResourceJobPreferenceService
      2. Method: GetResourceJobPreferenceResponseByResourceID() AddUpdateResourceJobPreferences(ResourceJobPreferenceRequest resourceJobPreferenceRequest)
   2. Save: Save data using transaction service
      1. Input parameter: ResourceJobPreferenceRequest
         1. This request parameter contains data related to jobpreferences and top most summary section.
      2. Output: ResourceJobPreferenceResponse
      3. Service: ResourceJobPreferenceTransactionalService
      4. Method Name: AddUpdateResourceJobPreferences
      5. namespace ZeroChaos.Service.TransactionalSevices
3. Work experience tab:
   1. Get: Load data using below service method.
      1. Service: ResourceJobPreferenceService
      2. Method: GetResourceJobPreferenceResponseByResourceID
   2. Save: Use below transaction layer service to save details.
      1. Input parameter: ResourceWorkExperienceRequest
      2. Output: ResourceWorkExperienceResponse
      3. Service: ResourceWorkExperienceTransactionalService
      4. Method Name: AddUpdateResourceWorkExperiencies
      5. namespace ZeroChaos.Service.TransactionalSevices
4. Skills & Certs tab:
   1. Get skills: Pull all skills from below service methods:
      1. Helper: ListPageHelper
      2. Method: GetAvailableCompetencies()
      3. namespace ZeroChaos.ZCWeb.Linq
   2. Get skills: Get selected resource skills using below service methods.
      1. Service: ResourceJobSkillService
      2. Method: GetResourceJobSkillByResourceID()
   3. Save skills: using below transaction service save summary and skills.
      1. Service: ResourceJobSkillTransactionalService
      2. Method: AddUpdateResourceJobSkills(List<ResourceJobSkill>)
   4. Get certs: pull all certification using below service methods:
      1. Helper: ListPageHelper
      2. Method: GetAvailableCompetencies()
      3. namespace ZeroChaos.ZCWeb.Linq
   5. Load certs: load selected certification:
      1. Service: ResourceJobSkillService
      2. Method: GetResourceJobSkillByResourceID()
   6. Save certs: save resource summary and certs:
      1. Service: ResourceCertificationTransactionalService
      2. Method: AddUpdateResourceCertifications(ResourceCertificationRequest)
5. Attachments tab:
   1. Resume: Load resume test using:
      1. Service: AttachmentMappingService
      2. Method: GetAttachmentMappingsByPKIDObjIDEventAttachmentID()
      3. Parms: resourceID, ObjectType.Resource, EventAttachmentName.RESUME\_AS\_PDF\_FILE
   2. Attachments: To load attachment we should call above(5.a) same service method twice with different eventAttachmentID
      1. For Resume: eventAttachmentID= RESUME\_AS\_PDF\_FILE
      2. For attachments: eventAttachmentID= Other
   3. Web references: Load all web references using below service method:
      1. Service: ResourceWebReferencesService
      2. Method: GetResourceWebReferencesByResourceID
   4. Save attachments and web references: use below transactionlayer:
      1. Service: ResourceResumeTransactionalService
      2. Method: AddUpdateResourceResume(ResourceResumeAttachmentRequest resourceResumeRequest)
      3. namespace ZeroChaos.Service.TransactionalSevices
6. Education tab:
   1. Load all entered education details:
      1. Service: ResourceEducationService
      2. Method: GetResourceEducationByResourceID()
   2. Save education details:
      1. Service: ResourceEducationTransactionalService
      2. Method: AddUpdateResourceEducation(ResourceEducationRequest)
      3. namespace ZeroChaos.Service.TransactionalSevices
   3. Load all saved language proficiencies:
      1. Service: ResourceLanguageProficiencyService
      2. Method: GetResourceLanguageProficiencyByResourceID
   4. Save language proficiency:
      1. Service: ResourceLanguageProficiencyTransactionalService
      2. Method: AddUpdateResourceLanguageProficiency (ResourceLanguageProficiencyRequest)
      3. namespace ZeroChaos.Service.TransactionalSevices
7. Profile Completeness section:
   1. Load Profile Completeness section:
      1. Service: ResourceService
      2. Method: GetProfileCompleteness(int resourceID)

**Interface Classes with Methods:**

1. Namespace : namespace **ZeroChaos.Service.Interfaces**
   1. Class: interface IResourceEducationService.

Methods:

/// This method will return ResourceEducation details from ResourceEducation table based on

ResourceID.

* + 1. List<ResourceEducation> GetResourceEducationByResourceID(int resourceID);

/// This method will insert the Resource Education details in ResourceEducation Table.

* + 1. int AddResourceEducation(ResourceEducation ResourceEducation);

/// This method will update the existing resource education record.

* + 1. bool UpdateResourceEducation(ResourceEducation ResourceEducation);

/// This method will populate the Education Tab of Resource Dossier.

/// This method will return ResourceEducationLanguageResponse which consist of ResourceEducation

/// Details and Resource Language Proficiency detail.

* + 1. ResourceEducationLanguageResponse GetResourceEducationLanguageResponseByResourceID (int resourceID);
  1. Class: interface IResourceJobClassificationService.

Methods:

/// This method will return List of Resource JobClassification based on ResourceID

* + 1. List<ResourceJobClassification> GetResourceJobClassificationByResourceId(int resourceID);

/// This method will insert Resource JobClassification based on ResourceID

* + 1. int AddResourceJobClassification(ResourceJobClassification resourceJobClassification);

/// This method will update the Resource Jobclassification

* + 1. bool UpdateResourceJobClassification(ResourceJobClassification resourceJobClassification);
  1. Class: interface IResourceJobPreferenceCountryService.

Methods:

/// This method will return the country names resource preferred to work or legally able to work.

* + 1. List<ResourceJobPreferenceCountry> GetResourceJobPreferenceCountryByJobPreferenceIDResourcePreferenceForCountryID(int jobPreferenceID, int resourcePreferenceForCountryID);

/// This method will add the country resource preferred to work.

* + 1. int AddResourceJobPreferenceCountry(ResourceJobPreferenceCountry resourceJobPreferenceCountry);

/// This method will update the country resource preferred to work.

* + 1. bool UpdateResourceJobPreferenceCountry(ResourceJobPreferenceCountry resourceJobPreferenceCountry);
  1. Class: interface IResourceJobPreferenceService.

Methods:

/// This method will return the resource job preferences details from ResourceJobPreference table

based on ResourceID

* + 1. ResourceJobPreference GetResourceJobPreferenceByResourceID(int resourceID);

/// This method will add the resource job preference details in ResourceJobPreference table.

* + 1. int AddResourceJobPreference(ResourceJobPreference resourceJobPreference);

/// This method will update the resource job preference details.

* + 1. bool UpdateResourceJobPreference(ResourceJobPreference resourceJobPreference);

/// This method will populate the Job Preference tab details of Resource Dossier.

/// This method will return ResourceJobPreferenceResponse which consist of Job Preference details

* + 1. ResourceJobPreferenceResponse GetResourceJobPreferenceResponseByResourceID(int resourceID);
  1. Class: interface IResourceJobSkillService.

Methods:

/// This method will return resource job skills details based on resourceid.

* + 1. List<ResourceJobSkill> GetResourceJobSkillByResourceID(int resourceID);

/// This method will add the resource job skill.

* + 1. int AddResourceJobSkill(ResourceJobSkill resourceJobSkill);

/// This method will update the resource job skill.

* + 1. bool UpdateResourceJobSkill(ResourceJobSkill resourceJobSkill);

/// This method will populate the skill and certifications tab details of Resource Dossier.

/// This method will return ResourceJobSkillCertificationResponse which consist of job skills details and

certification details of the resource.

* + 1. ResourceJobSkillCertificationResponse GetResourceSkillCertificationResponseByResourceID(int resourceID);
  1. Class: interface IResourceLanguageProficiencyService.

Methods:

/// This method will return resource language proficiency details based on resourceid.

* + 1. List<ResourceLanguageProficiency> GetResourceLanguageProficiencyByResourceID(int resourceID);

/// This method will add resource language proficiency details.

* + 1. int AddResourceLanguageProficiency(ResourceLanguageProficiency resourceLanguageProficiency);

/// This method will update resource language proficiency level.

* + 1. bool UpdateResourceLanguageProficiency(ResourceLanguageProficiency resourceLanguageProficiency);
  1. Class: interface IResourceService.

Methods:

/// This method will populate the My Dossier tab of Resource Dossier.

/// This method will return ResourceDossierResponse which consist of resource details and

personal information details of the resource.

* + 1. ResourceDossierResponse GetResourceDossierResponseByResourceID(int resourceID);

/// This method will populate the Requisition Summary tab of Resource Dossier.

/// This method will return ResourceRequisitionSubmissionResponse which consist of req submission

for the resource.

* + 1. ResourceRequisitionSubmissionResponse GetResourceRequisitionSubmissionResponseByResourceID(int resourceID);

/// This method will populate the Resume & Attachment tab of Resource Dossier.

/// This method will return ResourceResumeAttachmentResponse which consist of resume details and

attachment details for the resource.

* + 1. ResourceResumeAttachmentResponse GetResourceResumeAttachmentResponseByResourceID(int resourceID);

/// This method will populate the Personal Information tab of Resource Dossier.

/// This method will return ResourcePersonalInformationResponse which consist of resource personal

information details.

* + 1. ResourcePersonalInformationResponse GetResourcePersonalInformationResponseByResourceID(int resourceID);

/// This method will populate the Projects & Reviews tab of Resource Dossier.

/// This method will return ResourceProjectReviewResponse which consist of resource project details

and survey review details.

* + 1. ResourceProjectReviewResponse GetResourceProjectReviewResponseByResourceID(int resourceID);

/// This method will add / update the resource personal information.

* + 1. int AddUpdateResourcePersonalInformationByResourceID(ResourceRequest resourceRequest);

//// This method will get the Profile Completeness section of Edit dossier tab.

* + 1. Dictionary<String, String> GetProfileCompleteness(int resourceID);
  1. Class: interface IResourceWebReferencesService.

Methods:

/// This method return Web references details of the resource.

* + 1. List<ResourceWebReferences> GetResourceWebReferencesByResourceID(int resourceID);

/// This method add the Web references details of the resource.

* + 1. int AddResourceWebReferences(ResourceWebReferences resourceWebReferences);

/// This method update the Web references details of the resource.

* + 1. bool UpdateResourceWebReferences(ResourceWebReferences resourceWebReferences);
  1. Class: interface IResourceWorkExperienceService.

Methods:

/// This method return resource work experience details based on resourceid.

* + 1. List<ResourceWorkExperience> GetResourceWorkExperienceByResourceID(int resourceID);

/// This method will populate the Work experience tab of resource dossier

* + 1. ResourceWorkExperienceResponse GetResourceWorkExperiencesResponseByResourceID(int resourceID);

/// This method add resource work experience details.

* + 1. int AddResourceWorkExperience(ResourceWorkExperience resourceWorkExperience);

/// This method update resource work experience details.

* + 1. bool UpdateResourceWorkExperience(ResourceWorkExperience resourceWorkExperience);

/// This method will return all the work preference from look up table.

* + 1. List<WorkPreference> GetAllWorkPreferences();

1. Namespace : namespace **ZeroChaos.Service.TransactionalInterfaces**
   1. Class: interface IResourceCertificationTransactionalService.

Methods:

/// This method will add / update the resource certification details.

* + 1. bool AddUpdateResourceCertifications(ResourceJobSkillRequest resourceJobSkillRequest);
  1. Class: interface IResourceEducationTransactionalService.

Methods:

/// This method will add / update the resource education details.

* + 1. bool AddUpdateResourceEducations(ResourceEducationRequest resourceEducationRequest);
  1. Class: interface IResourceJobPreferenceTransactionalService.

Methods:

/// This method will add / update the resource job preference details.

* + 1. ResourceJobPreferenceResponse AddUpdateResourceJobPreferences(ResourceJobPreferenceRequest resourceJobPreferenceRequest);
  1. Class: interface IResourceJobSkillTransactionalService.

Methods:

/// This method will add / update the resource job skill details.

* + 1. bool AddUpdateResourceJobSkills(List<ResourceJobSkill> resourceJobSkills);
  1. Class: interface IResourceLanguageProficiencyTransactionalService.

Methods:

/// This method will add / update the resource language proficiency details.

* + 1. bool AddUpdateResourceLanguageProficiencies(List<ResourceLanguageProficiency> resourceLanguageProficiencies);
  1. Class: interface IResourceResumeTransactionalService.

Methods:

/// This method will add / update the resource Resume Attachment details.

* + 1. ResourceResumeAttachmentResponse AddUpdateResourceResumes(ResourceResumeAttachmentRequest resourceResumeAttachmentRequest);
  1. Class: interface IResourceWorkExperienceTransactionalService.

Methods:

/// This method will add / update the resource Work Experience details.

* + 1. ResourceWorkExperienceResponse AddUpdateResourceWorkExperiencies(ResourceWorkExperienceRequest resourceWorkExperienceRequest);

**Existing service methods to load data:**

1. Consume below service method to load data from master translations by passing corresponding objectid.
   1. Objects:
      1. Phone Type: objectID - 312
      2. Countries: objectID - 19
      3. Citizenship country: objectID - 19
      4. Preferred Language: objectID - 20
      5. Industry: objectID - 43
      6. EducationLevel: objectID - 7
      7. Currency: objectID - 18
   2. class: MasterTranslationService
      1. namespace ZeroChaos.Service.Services
   3. method: GetMasterTranslations(objectID, clientID, isDefaultEntryRequired, loggedOnUser, isActive)
2. States: Consume below service method to load data to states dropdown:
   1. class: CountryStateService
      1. namespace ZeroChaos.Service.Services
   2. method: GetCountryStatesByCountryCode(countryCode)
3. Preferred time zone:
   1. class: TimeZoneService
      1. namespace ZeroChaos.Service.Services
   2. method: GetTimeZonesByClientID(clientid)
4. Security Questions:
   1. class: SecurityQuestionService
      1. namespace ZeroChaos.Service.Services
   2. method: GetSecurityQuestions()
5. Update Security Questions:
   1. class: MembershipService()
      1. namespace ZeroChaos.Service.Services
   2. method: UpdateSecurityQuestionByUserID(membership)
6. Competencies:
   1. class: ListPageHelper
      1. namespace ZeroChaos.ZCWeb.Linq
   2. method: GetAvailableCompetencies(isSearchForFilter, startsWith, existingCompetencyIDs, out count)
7. Skill level:
   1. class: JobSkillService
      1. namespace ZeroChaos.Service.Services
   2. method: GetComeptencyLevelsForDropdown(0)
8. Months:
   1. class: ReqCandidateService
      1. namespace ZeroChaos.Service.Services
   2. method: GetMonths(cultureName)
9. Days:
   1. class: ReqCandidateService
      1. namespace ZeroChaos.Service.Services
   2. method: GetDays()
10. Update Password:
    1. class: MembershipService
       1. namespace ZeroChaos.Service.Services
    2. Method: ChangePasswordByUserID(membership)

**Existing partial views:**

1. To display skills use Requisition/\_ManageCompetenciesPartial.cshtml
2. To display all skills in popup use ‘\_AvailableCompetenciesPopup.cshtml’
3. For certification use ‘\_ManageCandidateCertifications.cshtml’
4. open all certification in popup using ‘\_AvailableCertificationsPopup.cshtml’
5. To display resume text use ‘\_ResumeDetailsPartial.cshtml’
6. To display attachment use ‘\_MultiAttachment.cshtml’

**Response Entities:**

Namespace: **ZeroChaos.Business.Entities**

* 1. Class: ResourceDossierResponse
     1. Resource Resource
     2. ResourceEducation ResourceEducation
     3. List<ResourceLanguageProficiency> ResourceLanguageProficiency
     4. ResourceSummary Summary
  2. class ResourceSummary:
     1. int ActiveProjectCount
     2. int PastProjectCount
     3. int TotalSubmission
     4. int Reviews
     5. List<string> Skills
     6. Dictionary<string, string> WorkExperiences
  3. Class: ResourceEducationLanguageResponse
     1. List<ResourceEducation> ResourceEducation
     2. List<ResourceLanguageProficiency> ResourceLanguageProficiency
  4. Class: ResourceJobPreferenceResponse
     1. int ResourceID
     2. string CareerGoal
     3. string AspiringQuotes
     4. bool ActionSucceded
     5. ResourceJobPreference ResourceJobPreference
     6. List<ResourceJobPreferenceCountry> ResourceJobPreferenceCountries
     7. List<ResourceJobClassification> ResourceJobClassifications
     8. List<ObjectIndustryMapping> ResourceIndustries
     9. List<ObjectJobCategoryMapping> ObjectJobCategoryMappings
     10. List<string> ResourceLegallyCountryToWork
     11. List<string> ResourcePreferredCountryToWork
  5. Class: ResourceJobSkillCertificationResponse
     1. List<ResourceJobSkill> ResourceJobSkills
     2. List<ResourceCertification> ResourceCertifications
  6. Class: ResourcePersonalInformationResponse
     1. Resource Resource
  7. Class: ResourceResumeAttachmentResponse
     1. List<ResourceResume> ResourceResume
     2. List<Attachment> ResourceAttachment
     3. List<ResourceWebReferences> ResourceWebReferences
     4. bool ActionSucceded
  8. Class: ResourceWorkExperienceResponse
     1. List<ResourceWorkExperience> ResourceWorkExperiences
     2. bool ActionSucceded
     3. List<Project> ProjectExperiences
  9. Class: ResourceProjectReviewResponse
     1. List<ResourceProjectReview> resourceProject

**Request Entities:**

Namespace: **ZeroChaos.Business.Entities**

1. Class: ResourceEducationRequest
   1. int ResourceID
   2. string CareerGoal
   3. string AspiringQuotes
   4. List<ResourceEducation> ResourceEducations
2. Class: ResourceJobPreferenceRequest
   1. int ResourceID
   2. public string CareerGoal
   3. string AspiringQuotes
   4. List<ResourceJobPreference> ResourceJobPreferences
3. Class: ResourceJobSkillRequest
   1. List<ResourceJobSkill> ResourceJobSkills
   2. int ResourceID
   3. string CareerGoal
   4. string AspiringQuotes
4. Class: ResourceCertificationRequest
   1. List<ResourceCertification> ResourceCertifications
   2. int ResourceID
   3. string CareerGoal
   4. string AspiringQuotes
5. Class: ResourceLanguageProficiencyRequest
   1. int ResourceID
   2. string CareerGoal
   3. string AspiringQuotes
   4. List<ResourceLanguageProficiency> ResourceLanguageProficiencies
6. Class: ResourceResumeAttachmentRequest
   1. int ResourceID
   2. string CareerGoal
   3. string AspiringQuotes
   4. ResourceResume ResourceResume
   5. List<object> ResourceAttachments
   6. List<ResourceWebReferences> ResourceWebReferences
7. Class: ResourceWorkExperienceRequest
   1. int ResourceID
   2. string CareerGoal
   3. string AspiringQuotes
   4. List<ResourceWorkExperience> ResourceWorkExperiences