****

**سند Technical سرویسهای Cloud**

|  |  |
| --- | --- |
| شناسه سند | DPV.AC.TEC |
| نگارش | 1.0 |
| تاريخ ارائه | 05/11/1394 |
| شرح سند | **اين سند، سند Technical زیرساخت مورد استفاده در پیاده سازی Cloud پروژه ی آناتولی ورانگر است. چنانچه مورد خاصی مبنی بر زیرساخت در تولید این پروژه مورد استفاده قرار گرفته باشد، به تفضیل در این سند مورد بررسی قرار خواهد گرفت. اين سند با ذکر عناوین تخصصی برنامه نویسی، در راستاي تكميل اسناد دیگر تهيه خواهد گرديد.** |

شركت داده کاوان پیشرو ایده ورانگر

اين سند جهت استفاده در شركت داده کاوان پیشرو ایده ورانگر تهيه شده است و هرگونه تكثير از اين سند در خارج از محدوده مورد توافق ممنوع مي‏باشد.

**سوابق نگارش و مرور**

|  |  |  |  |
| --- | --- | --- | --- |
| تاريخ | اقدام­كننده | سمت | شرح |
| 05/11/93 | مريم شاكري | تحليل**­**گر | تهيه نسخه پيش نويس |
|  | هومن احمدی | مدير پروژه |  |
|  | آرمان فضلی |  |  |

**فهرست مطالب**

Controllers

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | |  | |
| public IHttpActionResult GetUsers() | AccountsController | | Base | |
| public async Task<IHttpActionResult> GetUser(string Id) |
| public async Task<IHttpActionResult> GetUserByName(string username) |
| public async Task<IHttpActionResult> CreateUser(CreateUserBindingModel createUserModel) |
| public async Task<IHttpActionResult> ConfirmEmail(string userId = "", string code = "") |
| public async Task<IHttpActionResult> ChangePassword(ChangePasswordBindingModel model) |
| public async Task<IHttpActionResult> DeleteUser(string id) |
| public async Task<IHttpActionResult> AssignRolesToUser([FromUri] string id, [FromBody] string[] rolesToAssign) |
| public async Task<IHttpActionResult> AssignClaimsToUser([FromUri] string id, [FromBody] List<ClaimBindingModel> claimsToAssign) |
| public async Task<IHttpActionResult> RemoveClaimsFromUser([FromUri] string id, [FromBody] List<ClaimBindingModel> claimsToRemove) |
| protected static readonly log4net.ILog log = log4net.LogManager.GetLogger(System.Reflection.MethodBase.GetCurrentMethod().DeclaringType); | BaseApiController | |
| private ModelFactory \_modelFactory |
| private ApplicationUserManager \_AppUserManager = null; |
| private ApplicationRoleManager \_AppRoleManager = null; |
| protected ApplicationUserManager AppUserManager |
| protected ApplicationRoleManager AppRoleManager |
| public BaseApiController() |
| protected ModelFactory TheModelFactory |
| protected IHttpActionResult GetErrorResult(ModelStateDictionary modelState) |
| protected IHttpActionResult GetErrorResult(string error) |
| protected IHttpActionResult GetErrorResult(Exception ex) |
| public static void WriteExceptionDetails(Exception exception, StringBuilder builderToFill, int level, ModelStateDictionary modelState) |
| protected IHttpActionResult GetErrorResult(IdentityResult result) |
| public IHttpActionResult GetClaims() | ClaimsController | |
| public IHttpActionResult RefundOrder([FromUri]string orderId) | OrdersController | |
| public IHttpActionResult Get() |
| public async Task<IHttpActionResult> GetRole(string Id) | RolesController | |
| public IHttpActionResult GetAllRoles() |  | |
| public async Task<IHttpActionResult> Create(CreateRoleBindingModel model) |
| public async Task<IHttpActionResult> DeleteRole(string Id) |
| public async Task<IHttpActionResult> ManageUsersInRole(UsersInRoleModel model) |
| public class RequestModel | BaseTypeInfoController | BaseData | | Gateway | |
| public string privateOwnerId { get; set; } |
| public async Task<IHttpActionResult> GetBaseTypes(string privateOwnerId) |
| public async Task<IHttpActionResult> GetBaseTypes(string privateOwnerId, string dateAfter) |
| public async Task<IHttpActionResult> SaveBaseTypes(string privateOwnerId, List<BaseTypeViewModel> data) |
| public async Task<IHttpActionResult> GetReorderCalcTypes([FromBody] RequestModel data) |
| public async Task<IHttpActionResult> GetStockTypes(string privateOwnerId) |
| public async Task<IHttpActionResult> GetStockRequestTypes(string privateOwnerId) |
| public async Task<IHttpActionResult> SaveStockRequestTypes(string privateOwnerId, List<StockProductRequestTypeViewModel> data) |
| public async Task<IHttpActionResult> GetFiscalYears(string privateOwnerId) | FiscalYearController |
| public async Task<IHttpActionResult> GetFiscalYears(string privateOwnerId, string dateAfter) |
| public async Task<IHttpActionResult> SaveFiscalYears(string privateOwnerId, List<FiscalYearViewModel> data) |
| private static readonly log4net.ILog log = log4net.LogManager.GetLogger(System.Reflection.MethodBase.GetCurrentMethod().DeclaringType); | ManufactureController |
| public async Task<IHttpActionResult> GetManufactures(string privateOwnerId) |
| public async Task<IHttpActionResult> GetManufactures(string privateOwnerId, string dateAfter) |
| public async Task<IHttpActionResult> SaveManufactures(string privateOwnerId, List<ManufactureViewModel> data) |
| public async Task<IHttpActionResult> GetCityRegion(string privateOwnerId) | RegionController |
| public async Task<IHttpActionResult> GetCityRegion(string privateOwnerId, string dateAfter) |
| public async Task<IHttpActionResult> SaveCityRegionInfo(string privateOwnerId, List<CityRegionViewModel> data) |
| public async Task<IHttpActionResult> GetSuppliers(string privateOwnerId) | SupplierController |
| public async Task<IHttpActionResult> GetSuppliers(string privateOwnerId, string dateAfter) |  | |
| public async Task<IHttpActionResult> SaveSuppliers(string privateOwnerId, List<SupplierViewModel> data) |  | |
| public async Task<IHttpActionResult> GetCustomerBasketByCustomerId(string privateOwnerId, string customerId) | BasketController | Customer | |
| public async Task<IHttpActionResult> GetCustomerBasketByBasketId(string privateOwnerId, string basketId) |
| public async Task<IHttpActionResult> SaveBasket(string privateOwnerId, List<BasketViewModel> data) |
| public async Task<IHttpActionResult> DeleteBasket(string privateOwnerId, List<BasketViewModel> data) |
| public async Task<IHttpActionResult> SaveBasketItem(string privateOwnerId, List<BasketItemViewModel> data) |
| public async Task<IHttpActionResult> ChangeBasketItem(string privateOwnerId, List<BasketItemViewModel> data) |
| public async Task<IHttpActionResult> DeleteBasketitem(string privateOwnerId, List<BasketItemViewModel> data) |
| public async Task<IHttpActionResult> GetBasketitemByIds(string privateOwnerId, List<BasketItemViewModel> data) |
| public async Task<IHttpActionResult> GetCustomerById(string privateOwnerId, string id) | CustomerController |
| public async Task<IHttpActionResult> SaveCustomer(string privateOwnerId, CustomerViewModel data) |  |  | |
| public async Task<IHttpActionResult> GetActiveCustomerShipAddressByCustomerId(string privateOwnerId, string customerId) |  |  | |
| public async Task<IHttpActionResult> GetDefaultCustomerShipAddressByCustomerId(string privateOwnerId, string customerId) |  |  | |
| public async Task<IHttpActionResult> GetCustomerShipAddressByCustomerId(string privateOwnerId, string customerId) |  |  | |
| public async Task<IHttpActionResult> GetCustomerShipAddressByCustomerIdByLevel4(string privateOwnerId, string customerId, string regionId) |  |  | |
| public async Task<IHttpActionResult> SaveCustomer(string privateOwnerId, CustomerShipAddressViewModel data) |  |  | |
| private static readonly log4net.ILog log = log4net.LogManager.GetLogger(System.Reflection.MethodBase.GetCurrentMethod().DeclaringType); | ProductController | Product | |
| public async Task<IHttpActionResult> GetCharGroups(string privateOwnerId) |
| public async Task<IHttpActionResult> GetCharGroups(string privateOwnerId, string dateAfter) |
| public async Task<IHttpActionResult> SaveCharGroups(string privateOwnerId, List<CharGroupViewModel> data) |
| public async Task<IHttpActionResult> GetCharTypes(string privateOwnerId) |
| public async Task<IHttpActionResult> GetCharTypes(string privateOwnerId, string dateAfter) |
| public async Task<IHttpActionResult> SaveCharTypes(string privateOwnerId, List<CharTypeViewModel> data) |
| public async Task<IHttpActionResult> GetProducts(string privateOwnerId) |
| public async Task<IHttpActionResult> GetProducts(string privateOwnerId, string dateAfter) |
| public async Task<IHttpActionResult> SaveProducts(string privateOwnerId, List<ProductViewModel> data) |
| public async Task<IHttpActionResult> GetProductGroups(string privateOwnerId) |
| public async Task<IHttpActionResult> GetProductGroups(string privateOwnerId, string dateAfter) |
| public async Task<IHttpActionResult> SaveProductGroups(string privateOwnerId, List<ProductGroupViewModel> data) |
| public async Task<IHttpActionResult> GetMainProductGroups(string privateOwnerId) |  |
| public async Task<IHttpActionResult> GetMainProductGroups(string privateOwnerId, string dateAfter) |
| public async Task<IHttpActionResult> SaveMainProductGroups(string privateOwnerId, List<MainProductGroupViewModel> data) |
| private static readonly log4net.ILog log = log4net.LogManager.GetLogger(System.Reflection.MethodBase.GetCurrentMethod().DeclaringType); | ProductRateController |
| public async Task<IHttpActionResult> GetProductRates(string privateOwnerId) |  | |
| public async Task<IHttpActionResult> GetProductRateAvgs(string privateOwnerId) |  | |
| public async Task<IHttpActionResult> GetProductRates(string privateOwnerId, string dateAfter) |  | |
| public async Task<IHttpActionResult> SaveProductRates(string privateOwnerId, List<ProductRateViewModel> data) |  | |
| public async Task<IHttpActionResult> GetByCustomerId(string privateOwnerId, string customerId) | IncompletePurchaseOrderController | PurchaseOrder | |
| public async Task<IHttpActionResult> SaveIncompletePurchaseOrder(string privateOwnerId, List<IncompletePurchaseOrderViewModel> data) |
| public async Task<IHttpActionResult> ClearIncompletePurchaseOrder(string privateOwnerId, List<IncompletePurchaseOrderViewModel> data) |
| public async Task<IHttpActionResult> SaveIncompletePurchaseOrderItem(string privateOwnerId, List<IncompletePurchaseOrderLineItemViewModel> data) |
| public async Task<IHttpActionResult> ChangeIncompletePurchaseOrderItem(string privateOwnerId, List<IncompletePurchaseOrderLineItemViewModel> data) |
| public async Task<IHttpActionResult> DeleteIncompletePurchaseOrderitem(string privateOwnerId, List<IncompletePurchaseOrderLineItemViewModel> data) |
| public async Task<IHttpActionResult> CreateOrder(string privateOwnerId, PurchaseOrderViewModel orderEntity) | PurchaseOrderController |  | |
| public async Task<IHttpActionResult> CreateOrderLocal(string privateOwnerId, PurchaseOrderViewModel orderEntity) |  | |
| public async Task<IHttpActionResult> ClacPromo(string privateOwnerId, PurchaseOrderViewModel orderEntity) |  | |
| public async Task<IHttpActionResult> ClacPromoLocal(string privateOwnerId, PurchaseOrderViewModel orderEntity) |  | |
| public async Task<IHttpActionResult> GetPurchaseOrderOnlineByCustomerId(string privateOwnerId, string customerId) |  | |
| public async Task<IHttpActionResult> GetPurchaseOrderLocalByCustomerId(string customerId, string centerId) |  | |
| public async Task<IHttpActionResult> GetPurchaseOrderLineItemOnline(string privateOwnerId, string poId) |  | |
| public async Task<IHttpActionResult> GetPurchaseOrderLineItemLocal(string poId, string centerId) |  | |
| public async Task<IHttpActionResult> GetPurchaseOrderStatusHistoryOnline(string privateOwnerId, string poId) |  | |
| public async Task<IHttpActionResult> GetPurchaseOrderStatusHistoryLocal(string poId, string centerId) |  |  | |
| public async Task<IHttpActionResult> GetStockOnhands(string privateOwnerId) | StockController | Stock | |
| public async Task<IHttpActionResult> GetStockOnhands(string privateOwnerId, string dateAfter) |
| public async Task<IHttpActionResult> GetStockOnhandsByStockId(string privateOwnerId, string id) |
| public async Task<IHttpActionResult> SaveStockOnhands(string privateOwnerId, List<StoreActiveOnhandViewModel> data) |
| public async Task<IHttpActionResult> GetStocks(string privateOwnerId) |
| public async Task<IHttpActionResult> GetUserStocks([FromBody] RequestModel data) |
| public async Task<IHttpActionResult> SaveUserStocks([FromBody] RequestModel data) |
| public async Task<IHttpActionResult> GetStockCompletes(string privateOwnerId, string stockId) |
| public async Task<IHttpActionResult> GetStocks(string privateOwnerId, string dateAfter) |
| public async Task<IHttpActionResult> SaveStocks(string privateOwnerId, List<StockViewModel> data) |
| public async Task<IHttpActionResult> GetStockProducts(string privateOwnerId) |
| public async Task<IHttpActionResult> GetStockProductsByStockId([FromBody] RequestModel data) |
| public async Task<IHttpActionResult> GetStockProducts([FromBody] RequestModel data) |
| public async Task<IHttpActionResult> SaveStockProducts(string privateOwnerId, List<StockProductViewModel> data) |  |
| public class RequestModel | StockProductRequestController |
| public async Task<IHttpActionResult> GetStockProductRequestRules([FromBody] RequestModel data) |  | |
| public async Task<IHttpActionResult> GetStockProductRequestValidRules([FromBody] RequestModel data) |
| public async Task<IHttpActionResult> GetStockProductRequestRulesAfter([FromBody] RequestModel data) |
| public async Task<IHttpActionResult> SaveStockProductRequestRules(string privateOwnerId, List<StockProductRequestRuleViewModel> data) |
| public async Task<IHttpActionResult> GetStockProductRequestTypes([FromBody] RequestModel data) |
| public async Task<IHttpActionResult> SaveStockProductRequestTypes(string privateOwnerId, List<StockProductRequestTypeViewModel> data) |
| public async Task<IHttpActionResult> SaveStockProductRequests(string privateOwnerId, List<StockProductRequestViewModel> data) |
| public async Task<IHttpActionResult> GetStoreOnhands(string privateOwnerId) | StoreController | Store | |
| public async Task<IHttpActionResult> GetStoreOnhands(string privateOwnerId, string dateAfter) |  | |
| public async Task<IHttpActionResult> GetStoreOnhandsByStoreId(string privateOwnerId, string id) |
| public async Task<IHttpActionResult> GetStoreOnlineOnhandsByStoreId(string privateOwnerId, string id) |
| public async Task<IHttpActionResult> GetStoreLocalOnhandsByStoreId(string id) |
| public async Task<IHttpActionResult> GetStoreOnhandsAfter(string privateOwnerId, string id, string dateAfter) |
| public async Task<IHttpActionResult> SaveStoreOnhands(string privateOwnerId, List<StoreActiveOnhandViewModel> data) |
| public async Task<IHttpActionResult> GetStorePriceLists(string privateOwnerId) |
| public async Task<IHttpActionResult> GetStorePriceLists(string privateOwnerId, string id) |
| public async Task<IHttpActionResult> GetStorePriceListsAfter(string privateOwnerId, string dateAfter) |
| public async Task<IHttpActionResult> GetStorePriceListsAfter(string privateOwnerId, string id, string dateAfter) |
| public async Task<IHttpActionResult> SavePriceLists(string privateOwnerId, List<StoreActivePriceListViewModel> data) |
| public async Task<IHttpActionResult> GetStores(string privateOwnerId) |
| public async Task<IHttpActionResult> GetStores(string privateOwnerId, string dateAfter) |
| public async Task<IHttpActionResult> SaveStores(string privateOwnerId, List<StoreViewModel> data) |
| public async Task<IHttpActionResult> GetStoreCalendars(string privateOwnerId, string id) |
| public async Task<IHttpActionResult> SaveStoreCalendars(string privateOwnerId, List<StoreCalendarViewModel> data) |
| private static readonly log4net.ILog log = log4net.LogManager.GetLogger(System.Reflection.MethodBase.GetCurrentMethod().DeclaringType); | ImageManagerController | | ImageManager | |
| public IFileManager FileManager { get; set; } |
| public ImageManagerController() |
| public ImageManagerController(IFileManager fileManager) |
| public async Task<IHttpActionResult> GetProducts(string privateOwnerId) |
| public async Task<IHttpActionResult> GetProducts(string privateOwnerId, string dateAfter) |
| public async Task<string> SaveImage(string token, string imagetype, string privateOwnerId, string imageId) |
| public IHttpActionResult GetSample() | TestAuthController | | TestAuthController | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | |  |
| 1 | public IHttpActionResult GetUsers() | AccountsController | | Base |
| 2 | public async Task<IHttpActionResult> GetUser(string Id) |
| 3 | public async Task<IHttpActionResult> GetUserByName(string username) |
| 4 | public async Task<IHttpActionResult> CreateUser(CreateUserBindingModel createUserModel) |
| 5 | public async Task<IHttpActionResult> ConfirmEmail(string userId = "", string code = "") |
| 6 | public async Task<IHttpActionResult> ChangePassword(ChangePasswordBindingModel model) |
| 7 | public async Task<IHttpActionResult> DeleteUser(string id) |
| 8 | public async Task<IHttpActionResult> AssignRolesToUser([FromUri] string id, [FromBody] string[] rolesToAssign) |
| 9 | public async Task<IHttpActionResult> AssignClaimsToUser([FromUri] string id, [FromBody] List<ClaimBindingModel> claimsToAssign) |
| 10 | public async Task<IHttpActionResult> RemoveClaimsFromUser([FromUri] string id, [FromBody] List<ClaimBindingModel> claimsToRemove) |
| 11 | protected static readonly log4net.ILog log = log4net.LogManager.GetLogger(System.Reflection.MethodBase.GetCurrentMethod().DeclaringType); | BaseApiController | |
| 12 | private ModelFactory \_modelFactory |
| 13 | private ApplicationUserManager \_AppUserManager = null; |
| 14 | private ApplicationRoleManager \_AppRoleManager = null; |
| 15 | protected ApplicationUserManager AppUserManager |
| 16 | protected ApplicationRoleManager AppRoleManager |
| 17 | public BaseApiController() |
| 18 | protected ModelFactory TheModelFactory |
| 19 | protected IHttpActionResult GetErrorResult(ModelStateDictionary modelState) |
| 20 | protected IHttpActionResult GetErrorResult(string error) |
| 21 | protected IHttpActionResult GetErrorResult(Exception ex) |
| 22 | public static void WriteExceptionDetails(Exception exception, StringBuilder builderToFill, int level, ModelStateDictionary modelState) |
| 23 | protected IHttpActionResult GetErrorResult(IdentityResult result) |
| 24 | public IHttpActionResult GetClaims() | ClaimsController | |
| 25 | public IHttpActionResult RefundOrder([FromUri]string orderId) | OrdersController | |
| 26 | public IHttpActionResult Get() |
| 27 | public async Task<IHttpActionResult> GetRole(string Id) | RolesController | |
| 28 | public IHttpActionResult GetAllRoles() |
| 29 | public async Task<IHttpActionResult> Create(CreateRoleBindingModel model) |
| 30 | public async Task<IHttpActionResult> DeleteRole(string Id) |
| 31 | public async Task<IHttpActionResult> ManageUsersInRole(UsersInRoleModel model) |
| 32 | public class RequestModel | BaseTypeInfoController |  | Gateway |
| 33 | public string privateOwnerId { get; set; } |
| 34 | public async Task<IHttpActionResult> GetBaseTypes(string privateOwnerId) |
| 35 | public async Task<IHttpActionResult> GetBaseTypes(string privateOwnerId, string dateAfter) |
| 36 | public async Task<IHttpActionResult> SaveBaseTypes(string privateOwnerId, List<BaseTypeViewModel> data) |
| 37 | public async Task<IHttpActionResult> GetReorderCalcTypes([FromBody] RequestModel data) |
| 38 | public async Task<IHttpActionResult> GetStockTypes(string privateOwnerId) |
| 39 | public async Task<IHttpActionResult> GetStockRequestTypes(string privateOwnerId) |
| 40 | public async Task<IHttpActionResult> SaveStockRequestTypes(string privateOwnerId, List<StockProductRequestTypeViewModel> data) |
| 41 | public async Task<IHttpActionResult> GetFiscalYears(string privateOwnerId) |  |  |
| 42 | public async Task<IHttpActionResult> GetFiscalYears(string privateOwnerId, string dateAfter) |  |  |
| 43 | public async Task<IHttpActionResult> SaveFiscalYears(string privateOwnerId, List<FiscalYearViewModel> data) |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  | private static readonly log4net.ILog log = log4net.LogManager.GetLogger(System.Reflection.MethodBase.GetCurrentMethod().DeclaringType); |  | | ImageManager |
|  | public IFileManager FileManager { get; set; } |
|  | public ImageManagerController() |
|  | public ImageManagerController(IFileManager fileManager) |
|  | public async Task<IHttpActionResult> GetProducts(string privateOwnerId) |
|  | public async Task<IHttpActionResult> GetProducts(string privateOwnerId, string dateAfter) |
|  | public async Task<string> SaveImage(string token, string imagetype, string privateOwnerId, string imageId) |
|  | public IHttpActionResult GetSample() | TestAuthController | | TestAuthController |