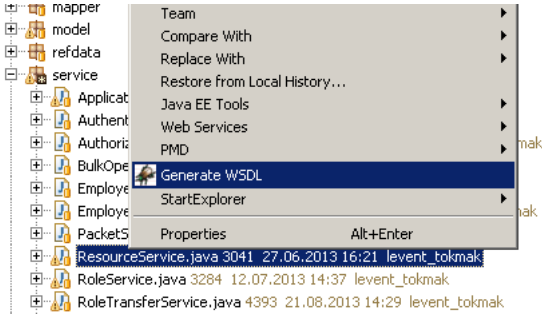
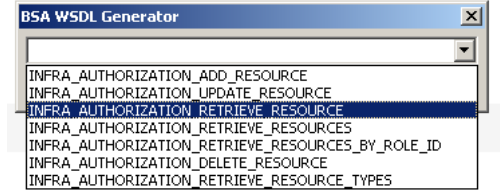


MCA Artifacts ve Wsdl oluşturma

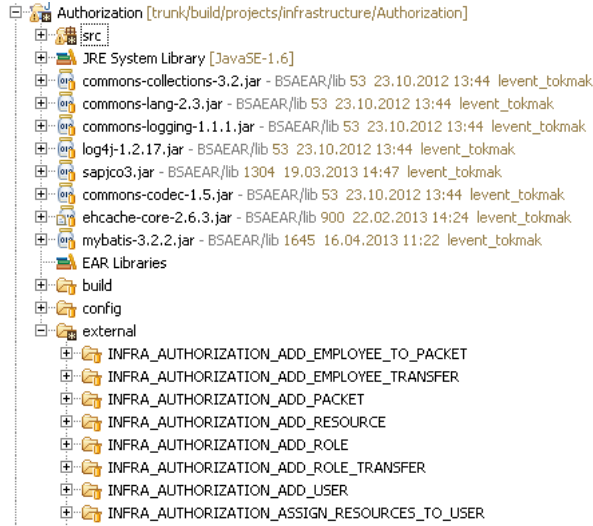
@External veya @Inbound annotation'ına sahip bir BSA servisi için MCA kodları ve wsdl otomatik olarak üretilir. Bunun için Eclipse'de "Project Explorer" view'inde herhangi bir service sınıfı üstünde sağ tıklayarak context menü açılır ve "Generate WSDL" ekranı açılır.



Açılan ekranda External servis listesi listelenir. İstenilen servis seçilip "Generate" butonuna tıklandığında dosyaları üretilir.



Üretilen dosyalar her projenin altında "external" dizinin altına servis adı ile oluşturulur.



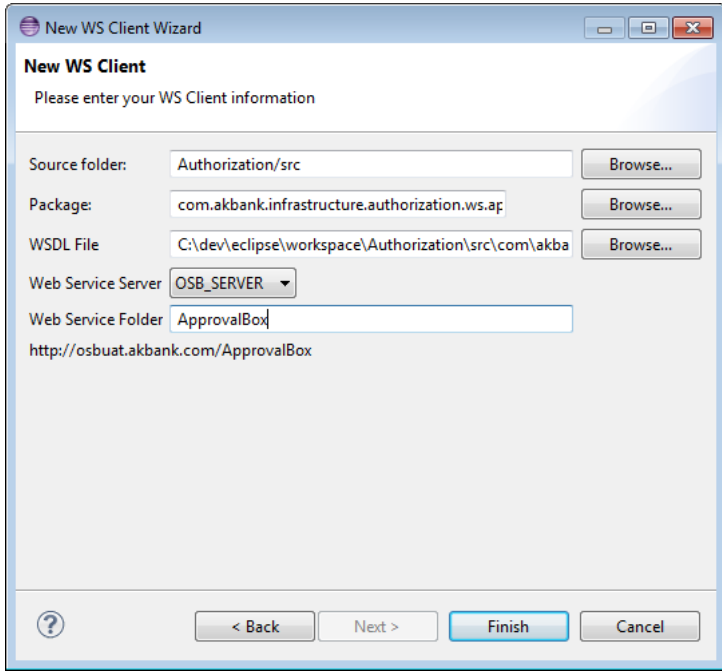
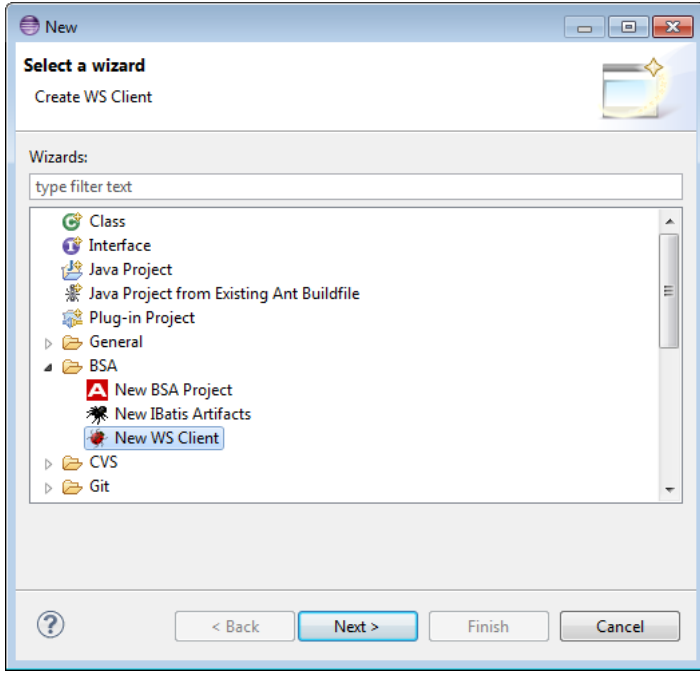
Web Servis çağırma

BSA üzerinden web servis çağırma için jaxws altyapısı kullanılmaktadır. Aşağıdaki adımlar takip edilerek web servis çağrılır.

1. com.akbank.[component name].[project name] altında ws.[service name] isimli bir package yaratılır,
2. Bu package üzerinde sağ click New\Other\New WS Client seçtikten sonra Web servis bilgileri girilir. Çağrılacak olan web servisin wsdl'i seçilir. **(File sistemden gösterilmelidir, url kullanıldığında hata alacaktır.)** Bağlanılacak server Web service Server combo'sundan seçilir. Eğer çağrılacak server combo'da yoksa BSA admin ekibinden talep edilir. Eğer Servis Web Service Server'ında bir klasör altında çalışıyorsa Web Service Folder alanı girilir. Finish butonuna basılınca jaxws client class'ları üretilir.
3. Oluşan client class'ları ile aşağıdaki şekilde web servis çağrılır.

```
ApprovalUserServices port = ApprovalUserServicesClient.getPort();
UserReturnDTO userReturn = port.matchBpmRoleWithAppRoles(bpmRole, appRoleIdsToAdd, appRoleIdsToDelete, usersToAdd, usersToDelete);
```

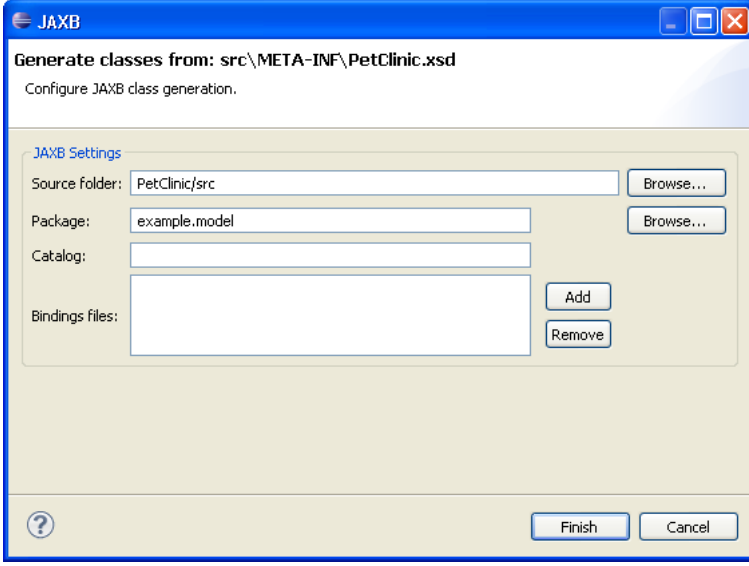
Not: WSDL dosyasında import edilen xsd'ler varsa com.akbank.[component name].[project name].ws.[service name] altına kopyalanmalıdır. WSDL'de geçen import ifadelerindeki url'ler silinip yerine kopyalanan xsd'lerin isimleri yazılmalıdır.



XSD'den Jaxb Class'ları oluşturma

BSA servislerinin input/output class'ları servisi çağırarak client ile aynı yapıda yapılmak isteniyorsa client xsd dosyası eclipse üzerinden import edilerek Jaxb class'ları oluşturulabilir ve oluşturulan class'ların servislerin input/output class'ları olarak kullanılabilir.

Navigator veya Project Explorer view'ında, .xsd dosyası üzerinde sağ tık,Generate > JAXB Classes menüsü seçilir. Aşağıdaki ekrandaki Jaxb class'larının bilgileri doldurulur.



Extending BSACache

Sık kullanılan, çok sık değişmeyen ve yüklemesi uzun süren datalar cache'e yüklenerek uygulama performansı artırılabilir. Cache'de data tutulması gereken durumlarda BSA Altyapı ekibi ile temase geçilmelidir. Cache'de tutulacak kayıt sayısı, cache'de objelerin ne kadar duracağı gibi tanımlamalar BSA altyapı ekibi tarafından echcache.xml dosyasında yapılır. Uygulama geliştirme ekipleri tarafından BSACache class'ını extend eden class'lar yazılır. Aşağıda override edilecek metodlar ve bu metodların işlevleri açıklanmıştır.

```
public class EmployeeCache extends BSACache<EmployeeCacheItem>{

    /**
     * CACHE ismini return eder
     */
    @Override
    public String getCacheName() {
        return "INFRA_AUTHORIZATION_EMPLOYEE_CACHE";
    }

    /**
     * Key değeri ile cache item yüklenir
     */
    @Override
    protected EmployeeCacheItem loadContent(String registrationNumber) {
        return EmployeeDao.retrieveExtendedEmployee(registrationNumber);
    }

    /**
     * Cache'de tutulacak tüm içeriği yükler
     */
    @Override
    public List<EmployeeCacheItem> loadAllContent() {
        return EmployeeDao.retrieveExtendedEmployees();
    }

    /**
     * Cache Item'i put eder. loadContent metoduna gönderilen parametre ile burada put edilen key değeri aynı olmalıdır.
     */
    @Override
    protected void put(EmployeeCacheItem content) {
        super.put(content.getRegistrationNumber(), content);
    }

    /**
     * Eğer server açılışında yüklenecekse loadOnStartup metodu override edilir. Default'da açılışta yüklenmez. İlk referansta yüklenir.
     */
    @Override
    public boolean loadOnStartup() {
        return true;
    }
}
```

Aşağıdaki kod kullanılarak cache instance'ına erişilebilir.

```
CacheFactory.getCache(EmployeeCache.class);
```

Reference Data

Farklı kaynaklardan erişilen kod/value tipinde lookup data'lara erişebilmek için referans data yapısı geliştirilmiştir. Referans data altyapısı ile lookup data'lara çok dilli olarak erişilebilmektedir. Referans data'lar cache data olarak sistemde tutulmakta ve günlük olarak yenilendiklerinden, referans data olarak erişilecek bilgiler çok nadir değişen bilgiler olmalıdır.

Defining new Reference Data

Yeni bir referans data tanımlamak için BSA projesindeki refdata package'ı altında RefDataLoader interface'ini implement eden bir class yaratıp, class'ın başına @RefData annotation tag'ini eklemek gerekmektedir. @RefData tag'i içerisine referans data'ya ulaşırken kullanılacak isim yazılır. RefDataLoader interface'inden gelen load metodu aşağıdaki örnekte olduğu gibi doldurulur. Örnekte veritabanından Application listesi getirilmektedir. İhtiyaç duyulması halinde web servis ile yada başka yöntemlerle verilere ulaşılabilir.

```
@RefData("AUTHORIZATION_APPLICATION_REFDATA")
public class ApplicationRefDataLoader implements RefDataLoader{

    @Override
    public RefDataContainer<?> load(String type, String language) {

        List<Application> list = ApplicationDao.selectAllApplications();

        Map<String, Application> data = new HashMap<String, Application>();
        for (Application application : list) {
            data.put(application.getId(), application);
        }

        RefDataImpl<Application> refDataImpl = new RefDataImpl<Application>(type, language, data);

        return refDataImpl;
    }
}
```

Retrieving Data from Reference Data

Referans data'ya aşağıdaki örnekteki gibi erişilebilir. Referans data'ya erişirken dil bilgisi göndermeye gerek yoktur. Referans data session'daki dile göre getirilir.

```
RefDataContainer<Application> applications = (RefDataContainer<Application>) ReferenceDataFactory.getReferenceData("AUTHORIZATION_APPLICATION_REFDATA");

List<Resource> resources = ResourceDao.retrieveResourcesAssignedToEmployee(null, registrationNumber);

for (Resource resource : resources) {
    outBag.put("RESOURCES", index, "ID", resource.getId());
    outBag.put("RESOURCES", index, "CODE", resource.getCode());
    outBag.put("RESOURCES", index, "NAME", resource.getName());
    outBag.put("RESOURCES", index, "DESCRIPTION", resource.getDescription());
    outBag.put("RESOURCES", index, "APP_ID", resource.getAppId());
    outBag.put("RESOURCES", index, "APP_CODE", applications.get(resource.getAppId()).getCode());
    outBag.put("RESOURCES", index, "APP_DESCRIPTION", applications.get(resource.getAppId()).getDescription());
    index++;
}
```

BSA_LOOKUP_DATA tanımlama

Genel lookup data ihtiyaçları için BSA_LOOKUP_DATA referans datası framework tarafından hazırlanmıştır. BSA_LOOKUPDATA tablosuna girilecek lookup data için herhangi bir loader class yazılmasına ihtiyaç yoktur. BSA_LOOKUPDATA tablosu aşağıdaki kolonlardan oluşur.GROUP_CODE kolonuna referans data ismi girilir.

- GROUP_CODE
- CODE
- LANGUAGE
- DESCRIPTION

Örnek olarak BSA_LOOKUPDATA tablosuna aşağıdaki kayıtların eklenmesi durumunda CITIES isimli bir referans data tanımlanmış oluyor.

GROUP_CODE	CODE	LANGUAGE	DESCRIPTION
CITIES	05	tr	Amasya
CITIES	23	tr	Elazığ

ReferenceDataFactory.getReferenceData metodu CITIES referans data kodu ile çağrılırsa şehir bilgilerine ulaşılabilir.

```
RefDataContainer<String> cities= (RefDataContainer<string>) ReferenceDataFactory.getReferenceData("CITIES");
cities.get("05");
```

Calling Service With ServiceExecutor

BSA servislerinden başka BSA servisleri ServiceExecutor class'ı kullanılarak çağrılır. BSA servisleri üç metodla çağrılabilir.

- execute** metodu aynı transaction içerisinde başka bir servis çağırmak için kullanılır. execute metoduna çağrılan servisin input key'lerini bulunduran bir bag gönderilir. Çağrılan servisin döndüğü bag nesnesini döndürür.

```
Bag usersBag = new Bag();
usersBag.put("USER_ID",userId);
Bag insertUserBag = ServiceExecutor.execute("INFRA_AUTHORIZATION_INSERT_USER", usersBag);
boolean isInserted = insertUserBag.get("IS_INSERTED").toBoolean();
```

- executeInNewTx** metodu execute metodu ile aynı şekilde çağrılır. Tek farkı executeInNewTx metodu ile çağrılan servis yeni bir transaction açılarak çağrılır. Bu şekilde bir servis içerisinde executeInNewTx metodu ile başka bir servis çağırdıktan sonra eğer servisimiz hata alsa bile çağırdığımız servis rollback olmaz.
- asyncExecute** metodu asenkron olarak servis çağırmak için kullanılır. Asenkron servis çağırma işleminde işlemin tamamlanmadığı kontrol edilmek istenirse replyMessage paramtereri true olarak gönderilir. Bu durumda asyncExecute metdou başlatılan thread için correlationId değeri döner.Başlatılan servisin durumu aşağıdaki şekilde kontrol edilir. Asenkron olarak çağrılan servis için yeni transaction başlatılır.

```
String correlationId = ServiceExecutor.asyncExecute("TEST_ASENKRON_REPLY", new Bag(), true);
Bag bag = JmsUtil.getMessage(correlationId);

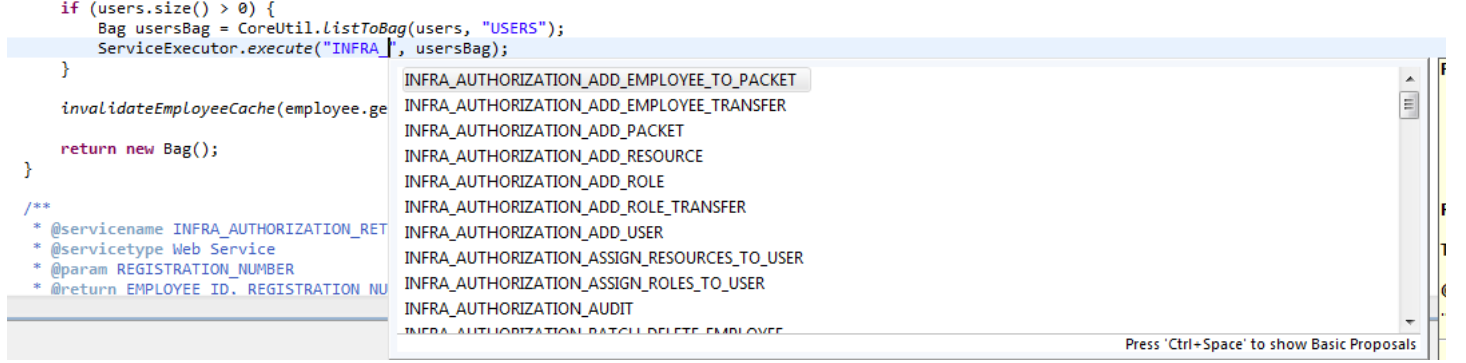
//Eğer bag==null ise servis tamamlanmamış olabilir. AsyncServiceExecutor.getMessage metodunun timeour değeri 1 saniyedir.
```

```

if (bag!=null){
    String status = bag.get("STATUS").toString();
    if (status.equals("SUCCESS")){
        Bag serviceOutBag = bag.get("SERVICE_OUTBAG").toBag();
    }else{
        ...
    }
}

```

Servis çağırırken servis ismine çabuk erişim için çift tırnak arasında Ctrl+Space tuşlarına basılırsa en uygun servis isimleri listelenir. Listelenen servislerin üzerine tıklanarak input/output parametreleri de görülebilir.



ConfigurationHelper kullanımı

BSA altyapısının ve BSA projelerinin kullandığı property'ler database'de BSA_CONFIG tablosunda tutulur. Örnek:

SECTION	TAG	VALUE	CLUSTER_NAME
EnvironmentConfig	Language	tr	ALL
EnvironmentConfig	mqEnabled	false	ALL
EnvironmentConfig	mqEnabled	true	10.215.75.220
EnvironmentConfig	OSB_PASSWORD	AUTH_pwd0651	ALL
EnvironmentConfig	OSB_PASSWORD_2	branchuser290	ALL
EnvironmentConfig	OSB_SERVER	https://akosbint.akbank.com.tr	ALL
EnvironmentConfig	OSB_SERVER_2	http://osbint.akbank.com	ALL
EnvironmentConfig	OSB_USER	authBoxUser	ALL
EnvironmentConfig	OSB_USER_2	SubeBMUser	ALL
EnvironmentConfig	quartz	true	ALL
EnvironmentConfig	serviceLogEnabled	false	ALL
EnvironmentConfig	SSLProvider	C:\keystore.jks	ALL
EnvironmentConfig	validateServiceParams	true	ALL
FtpConfig	FtpPassword	h2RJoOie	ALL
FtpConfig	FtpServerAddress	ftp.akbank.com.tr	ALL
FtpConfig	ftpUserName	FR4MUPLV	ALL
scheduledServices	INFRA_AUTHORIZATION_RELOAD_EMPLOYEE_CACHE	true	ALL
scheduledServices	INFRA_AUTHORIZATION_RELOAD_USER_CACHE	true	ALL

- **Override** edilmek istenen değer için yeni bir row oluşturulur ve CLUSTER_NAME kolonuna local makinenin IP adresi yazılır.
- **Yeni bir property** tanımlarken CLUSTER_NAME değeri ALL olarak eklenmelidir.
- BSA_CONFIG tablosu test ortamlarına manuel olarak taşınır.
- Property'lerden birinin değeri değiştirildiğinde aktif hale gelmesi için server'ların restart olması gerekir.

Değerlere ConfigurationHelper class'ındaki static metodlar çağrılarak erişilir.

- **Bir section altındaki tüm tag değerlerine erişmek için** : ConfigurationHelper.getConfiguration(String sectionName)
- **Bir section altındaki belirli bir tag değerine erişmek için** : ConfigurationHelper.getConfigurationValue(String sectionName, String tagName)

BSAContext

BSA uygulamasında bir thread boyunca taşınan session datasına **BSAContext.getInstance()** metodu çağrılarak erişilebilir. BSAContext objesi framework tarafından dışarıdan bir request geldiği zaman oluşturulur ve tüm servisler bittiği zaman temizlenir. BSAContext class'ında aşağıdaki değişkenler tutulmaktadır, ihtiyaca göre eklemeler yapılabilir.

- requestId
- customerId
- userId
- entityId
- terminal
- channel

- locale

Db Authorization

BSA'dan her bir db için bir uygulama kullanıcısı ile connection kurulmaktadır. Bir db'deki her kullanıcı kendi objelerine erişebilmesi için o db'deki uygulama kullanıcısına yetki vermektedir. Bu durum aynı db'de bulunan kullancılarının BSA üzerinden birbirlerinin verilerine erişebilmelerine olanak sağlamaktadır. Bu durumun etkisini en aza indirmek için BSA altyapısında db'ye atılan her sorgu incelenmekte ve uygulamanın erişmek istediği tablolara erişim yetkisinin olup olmadığı kontrol edilmektedir. Uygulamaların erişeceği tablo bilgileri BSA db mapping tablosunda tutulmaktadır. Tablo yetkisi tablo prefix'i ile ya da tablonun tam adı ile verilebilmektedir. Tablo erişimlerinde "DbAuthorization Error" hatası alındığında ilgili stacktrace ile birlikte BSA grubuna Jira issue'su açılması gerekmektedir. BSA projelerinde mapper xml'lerde zorunlu olmadıkça \$ yerine # kullanılması gerekmektedir. \$ kullanıldığında Mybatis string replacement yapmakta, bu nedenle \$ kullanıldığında \$'a geçilen farklı her parametre değeri için farklı bir sql oluşturulmaktadır. BSA altyapısı ve db'ler daha önce hiç çalışmamış olan sql'leri parse ettiğinden \$ kullanımı performans için olumsuz bir durum oluşturmaktadır. Zorunlu olmayan \$ kullanımlarının tespiti durumunda \$ kullanılan uygulamalarda kod değişikliği yapılması gerekecektir.

Service Authorization

BSA içinde ServiceExecutor üzerinden servisler birbirlerini çağırabilmektedir. Aynı proje içinde olmayan servislerin birbirlerini çağırabilmelerini kontrol altında tutmak amacıyla BSA altyapısında bir servis yetkilendirme geliştirmesi yapılmıştır. Aynı proje içindeki servisler birbirlerini bu kontrole takılmadan çağırmaya devam edebileceklerdir.

Servis yetkilendirmesi için önceden @Service annotation'ına allowedServices attribute'u eklenmekteydi. Bunun yerine artık bu bilgi db'de tutulmaktadır.

Bir servisin farklı projelerdeki servislerden çağırılabilmesi isteniyorsa db'de service authorization tablosunda bu servise ait bir kayıt oluşturulmalı ve izin verilecek servisler bu kayıta belirtilmelidir. İzin verilecek birden fazla servis varsa servis adları ; ile ayrılmalıdır.

İzin verilecek servis adına servis adı, proje adı, (varsa) subcomponent adı ya da component adı verilebilir. Servis adı verildiğinde sadece o servise, proje/subcomponent/component adı verildiğinde o proje/subcomponent/component altındaki tüm servislere izin verilir. Öncelik servis ya da proje adı verilmesi yönündedir. Subcomponent/component adı verildiği durumlar sorgulanacak, gerekli görülmezse buna izin verilmeyebilecektir.

Db'de service authorization tablosunda bir servise ait bir kayıt yoksa o servise farklı projelerden servisler erişemeyecektir.

Adding Third Party Libraries

Third Party kütüphaneler kullanılması gerektiğinde, ilgili jar dosyalarının pakete eklenmesi için BSA altyapı ekibi ile temasa geçilmelidir.

Kod Review Check List

BSA ortamında geliştirilen projelerin belirli kalite standartlarına uygun olarak geliştirilmeleri için PMD uygulaması kullanılmaktadır. SVN repository'sine commit edilmiş kodlar deployment paketine eklendikten sonra deployment script'i tarafından taranmakta ve PMD sonuçları BSADevelopers mail grubuna deployment sonuçları ile beraber gönderilmektedir. Ayrıca eclipse üzerinde workspace.support\qa altında build.properties dosyasında proje ismine kontrol edilmek istenen projenin ismi girilerek build.xml script'indeki view.report ant task'ı çalıştırılarak ta PMD raporu görüntülenebilir.

PMD ruleset ile detay açıklamaya <http://pmd.sourceforge.net/> adresinden erişilebilir.

PMD kural set'inde hatalar P1-P5 arasında seviyelendirilmiştir. P1 ve P2 hatalar kritik olup developer'lar tarafından düzeltilmelidir. **Üretime geçişe uygun olmayan projeler üretime ortamına deploy edilecek BSAEAR paketine eklenmeyecektir.**

BSA ortamındaki uygulamaların uyması gereken kritik seviyedeki kurallar aşağıda açıklanmıştır.

Java Standards

Project Dependency

BSA projeleri birbirlerine compile zamanında bağımlı olamazlar. Başka bir projedeki BSA servisi çağrılacaksa ServiceExecutor.execute metodu ile çağrılır.

```
ServiceExecutor.execute("INTEGRO_AKROBAT_COMMON_GET_FLAG_VALUE", inBag);
```

Servis isimlendirmeleri

Servis isimleri büyük harfle [COMPONENT_NAME]_[PROJECT_NAME]_[SERVIS_ISMI] olarak verilmelidir.

```
@Service("INTEGRO_MYNOTES_INSERT_NOTE")
```

Servis Javadoc

Servis metodlarına Javadoc eklenmelidir. Javadoc'larda aşağıdaki örnekteki gibi servisin input/output bag objelerindeki key isimleri belirtilmelidir.

```
/**
 * @param APP_ID
 * @param BRANCH_CODE
 * @param BRANCH_SATELLITE_NUMBER
 * @param REGISTRATION_NUMBER
 * @param STATUS
 * @return USERS {REGISTRATION_NUMBER ,FIRST_NAME, LAST_NAME, UNIT_CODE, PARENT_UNIT_CODE, JOB_LEVEL_CODE
 *                ,JOB_CODE, EMAIL, PHONE_NUMBER, REFERENCE_NUMBER, IS_SAP_DATA, REFERENCE_NUMBER}
 * @throws BSAException
 */
@Service("INFRA_AUTHORIZATION_RETRIEVE_USERS")
public static Bag retrieveUsers(Bag inBag) throws BSAException {
}
```

Fiziksel Delete

BSA ortamında yaratılan tablolar STATUS isimli bir kolon bulundurmaktadır. Delete işlemi yapılacağı zaman fiziksel delete yapmak yerine STATUS kolonunun değeri '0' olarak güncellenmelidir.

```
public static void delete(String id){
    Job j = mapperQuery.selectByPrimaryKey(id);
    j.setStatus("0");
    mapperQuery.updateByPrimaryKeySelective(j);
}
```

DAO katmanı

DAO class'larında sadece database işlemleri yapılmalıdır. Eğer database işlemlerinde kontroller yapılması gerekiyor ise bu kontroller utility olarak yapılmalıdır.

Try\Catch

Zorunlu durumlar dışında Servis yada DAO katmanlarında try\catch bloğu kullanılmamalıdır. Hata Loglaması, Exception yönetimi tarzı işler BSA framework tarafından yürütülmektedir.

SystemPrintln

Loglama işlemleri için System.out.println kullanılmamalıdır. Loglama log4j kullanılarak yapılmalıdır.

```
private final static Logger log = Logger.getLogger(Example.class);
log.info("Info Log....");
```

Metod uzunluğu

model ve mapper package altındaki class'lar dışında kalan class'larda metod uzunluğu 100 satırdan fazla olamaz.

Metod parametre sayısı

service ve dao package altındaki class'larda yaratılan metodlar en fazla 10 parametre alabilir.

Mapper Factory referans

MapperFactory class'ına sadece dao class'larından referans edilebilir.

Servis package

Servis class'ları service package'ı altında yaratılmalıdır.

Package isimlendirmeleri

Package isimleri com.akbank ile başlamalıdır.

Generic Exception catch etme

Generic exception'lar catch edilmemelidir.

Throwable catch etme

Throwable catch edilmemelidir.

BSA Ortam bilgileri

Uygulama sunucuları

DEV	Uygulama Sunucusu	172.21.131.54
	Uygulama adresi	http://bsadevelopment.akbank.com.tr (http://bsadevelopment.akbank.com.tr/)
INT	Uygulama Sunucusu	172.21.127.65
	Uygulama adresi	https://bsaint.akbank.com.tr
UAT	Uygulama Sunucusu	172.21.130.87
	Uygulama adresi	http://bsauat1.akbank.com.tr (http://bsauat1.akbank.com.tr/)
PREPROD	Uygulama Sunucusu	172.21.130.249 172.21.130.166
	Uygulama adresi - Venus	https://bsavenuspre.akbank.com.tr
	Uygulama adresi - Batch	https://bsabatchpre.akbank.com.tr
	Uygulama adresi - Merkur	https://bsamerkurpre.akbank.com.tr
	Uygulama adresi - DYS1	https://bsadyspre.akbank.com.tr
	Uygulama adresi - MKY	https://bsamkypre.akbank.com.tr
PROD	Merkur	https://bsamerkur.akbank.com.tr BSAPAP01 - 172.31.128.227 BSAPAP02 - 172.31.128.233
	Venus	https://bsavenus.akbank.com.tr BSAPAP01 - 172.31.128.227 BSAPAP02 - 172.31.128.233
	Batch	https://bsabatch.akbank.com.tr BSAPAP03 - 172.31.128.229 BSAPAP04 - 172.31.128.218
	Log	https://bsalog.akbank.com.tr BSAPAP03 - 172.31.128.229 BSAPAP04 - 172.31.128.218
	MKY	https://bsamky.akbank.com.tr BSAPAP05 - 172.31.128.226 BSAPAP06 - 172.31.129.79
	DYS	https://bsadys.akbank.com.tr BSAPAP07 - 172.31.129.80 BSAPAP08 - 172.31.129.81
	Investment	https://bsainvestment.akbank.com.tr

BSA test sunucularına test ortamlarında bsalog\bsalog123 kullanıcısı ile bağlanarak loglara erişilebilir.

PROD sunucularına erişmek için kullanıcı yaratılmalı ve Base plus yetkisi alınmalıdır.

Veritabanı kullanıcıları

Ortam	Uygulama kullanıcısı	BSA Core Schema (Kullanıcı adı / şifre)	Diğer
DEV	DBSAUSR1 TYR67_yds DBSAUSR2 YFM_88rsc	DBSA bbsaa_13	DIAM, DMID, DMTS, ... vb
INT	IBSAUSR1, IBSAUSR2	IBSA	IIAM, IMID, IMTS, ... vb
UAT	UBSAUSR1, UBSAUSR2	UBSA	UIAM, UMID, UMTS, ... vb
PROD	PBSAUSR1, PBSAUSR2	PBSA	PIAM, PMID, PMTS, ... vb

* Üretim ortamına aktarımlar talep ile yapılır

BSA ortamında uygulamalar tarafından erişilecek tablolar için sadece synonym oluşturulması amacıyla aşağıdaki kullanıcı ile login olunarak istenilen scriptler çalıştırılabilir.

Mapper class'lar BSA01Interface'i extend ediyorsa uygulama kullanıcısı *BSAUSR1, BSA02Intercafe'i extend ediyorsa uygulama kullanıcısı *BSAUSR2 kullanıcısını kullanmaktadır.

GRANT ve SYNONYM'leri extend edilen interface ismini göre uygun kullanıcıya vermek gerekmektedir.

INT ortamı :
Kullanıcı adı : IBSAUSR3
Şifre : Akbank_123

UAT ortamı :
Kullanıcı adı : UBSAUSR3
Şifre : Akbank_123

PREPROD ortamı :
Kullanıcı adı : RBSAUSR3
Şifre : HFT77zst

Örnek script :
create or replace SYNONYM IBSAUSR1.IAM_EMPLOYEE FOR IIAM.IAM_EMPLOYEE

Datasource'ların kullandığı kullanıcılar :

jndi name	DEV	INT	UAT	PREPROD	PROD
jdbc/DYSTAP2	ICMTCONC	ICMTCONC	ICMTCONC		
jdbc/KSCMTAP1	ICMCONCT	ICMCONCT	ICMCONCT		
jdbc/akbForeksDS	thisusr1	thisusr1	thisusr1		
jdbc/akbForeksDelayedDS	thisusr1	thisusr1	thisusr1		
jdbc/akbGeneksDS	BSATESTUSR1	BSATESTUSR1	BSATESTUSR1		
jdbc/akyGeneksDS	BSAUSR1	BSAUSR1	BSAUSR1		
jdbc/bsa01DS	DBSAUSR1	IBSAUSR1	UBSAUSR1		
jdbc/bsa02DS	DBSAUSR2	IBSAUSR2	UBSAUSR2		
jdbc/crmDS	TCRMBSAUSR1	TCRMBSAUSR1	TCRMBSAUSR1		
jdbc/customerAnalyticsDS	MMBBUSR1	MMBBUSR1	MMBBUSR1		
jdbc/dysArchiveDS	DDYSAUSR1	TDYSAUSR1	KDYSAUSR1		
jdbc/dysDS	DDYSUSR1	IDYSUSR1	UDYSUSR1		
jdbc/fraudDS	TFIMUSR1	TFIMUSR1	TFIMUSR1		
jdbc/hostDS	A525208	A525209	A525212	A525215	A525207
jdbc/mcaDS	TMCACOM	TMCACOM	TMCACOM		
jdbc/mobilDS	TKUM	TKUM	TKUM		
jdbc/oceanDS	FUYEUSR1	FUYEUSR1	FUYEUSR1		
jdbc/sysDS	tomnusr1	tomnusr1	tomnusr1		
jdbc/treasuryDS	DTREUSR1	ITREUSR1	UTREUSR1		

Deployment işlemleri

Geliştirilen BSA projesinin Meister tarafından build edilip INT, UAT ve PREPROD ortamlarına deploy edilmesi için öncelikle ilgili TGT dosyasının oluşturulması gerekir. Bu dosyalar BSA/build/tgt dizininde bulunur; varolan bir dosya kopyalanarak dosya adı ve içeriğinde gerekli değişiklikler yapılır ve commitlenir.

Bir projenin üretim ortamına gönderilmesi için öncelikle BSA ekibine aşağıdaki kategorilerde JIRA kaydı açılır. Bu JIRA kaydının açıklama kısmında üretim ortamına gönderilecek değişiklikleri içeren TFS changeset numaraları belirtilir. Bu işlem geçiş günü en geç saat 14:00'a kadar tamamlanmalıdır.

Project	Kurumsal Mimari
Issue Type	Task
Component	Banking Service Architecture

Description	TFS changeset numaraları
-------------	--------------------------

TFS changeset'leri taşındıktan sonra BSA ekibi build başlatır ve build sonrası oluşan JAR dosyalarının tümü aşağıdaki dizinlere kopyalanır.

\\Abmfs001\Deployment_STG\BSAPAP1-2

TFS changeset'lerinin taşınması için açılan JIRA kayıt örneği:

<https://jira.akbank.com/browse/KRM-3563?filter=14237>

Değişiklik yönetimine, üretim ortamına deployment için BT Aksiyon üzerinden kayıt açılır. Bu kaydın açılması için yukarıda bahsedilen dizine BSA ekibi tarafından dosyaların kopyalanmasının beklenmesine gerek yoktur. Değişiklik yönetimi ekibi tarafından dosyalar deployment sırasında alınacağı için kopyalama işleminden önce de DY kaydı açılabilir.

http://lotus2.akbank.com.tr/?OpenDocument&OpenType - Görev Formu - Internet Explorer

Deployment
Statü : **Tamamlandı** Yetkili :

▼ Talepteki Diğer Görevler :

Kayıd Açanın Bilgileri			
Kayıd Açan		48592-OSMAN KARAGÖZ	Kayıt T
Birim		9091-BT KURUMSAL MİMARİ YÖN. VE ALTYAPI UYG.	Talep
Alt Grup		BTSYUAG - SÜREÇ YÖNETİMİ UYGULAMALARI ALT GR	Ortam
Görev Tanımı			Katego
İmza Yetkisi			İdari C
Talep Açıklama		Merhaba, BSA MKY ortamında yeni sürümün devreye alınmasını rica ederim. İyi Çalışmalar	İdari C
Talep Ek Dokümanları			
Görev Açıklama		Merhaba, BSA MKY ortamında yeni sürümün devreye alınmasını rica ederim. İyi Çalışmalar	
Bu görevi bekleyen görevler			Talebe

İlgili Konfigürasyon Birimleri		Application/ In-House Server/ BSA/ IHS-2011-18
Plan Tipi		03 Ocak Haftalık Geçiş
Uygulama Tipi		BSA
İşlem Yapılacak Zaman		03.01.2017 22:00
İşlem Açıklaması		Deploy edildi.batch ve mky ortamına
Görev Ek Dokümanları		

Bu görev PROD ortamı için idari onaylı olup, idari onay için yöneticinize gidecektir. Görev Grubu DEĞİŞİKLİK YÖNETİMİ ALT grubudur.

Planlı geçişlerde yapılacak standart WAS deploymentları 22:00 den sonra yapılacaktır.

Deployment sonrası testler uygulama grubunun ve iş biriminin sorumluluğundadır.

Uygulama Adı	Bankacılık Servis Altyapısı(BSA)	İstenen tar
Deployment yapılacak serverlar	BSAPAP03 - BATCH - 172.31.128.229 BSAPAP04 - BATCH - 172.31.128.218 BSAPAP05 - MKY - 172.31.128.226 BSAPAP06 - MKY - 172.31.129.79	
Dosyanın bulunduğu path	\\abmfs001\Deployment_STG\BSAPAP1-2\library	Deploy edil
Yapılacak değişikliğin hizmete etkisi	Kesinti Yok	Sorun halin planı

Deployment Sonrası Kontroller

BSA uygulaması açılışında her JVM için startup.log dosyasında BSA paketinin versiyonu ve uygulamanın başladığına dair log atılır. Bu log mesajları atılmadı ise uygulama loglarından system-error.log ve Was logları(SystemOut, SystemErr) kontrol edilir. Uygulamanın düzgün açıldığı görüldükten sonra SOAP UT'dan paket içerisindeki servisler test edilebilir. Servisler test edilirken business hataları alınması durumunda service-error.log dosyasından kontrol edilir.

Uygulama açılışında startup.log örnek :

```
2014-07-22 22:43:46,252 INFO (ApplicationStartup.java:22) BSA Application version : i1.0 b10448 r-1
2014-07-22 22:43:51,806 INFO (ApplicationStartup.java:34) com.akbank.bsa.core.startup.SSLProviderInitializer - initializer starting
2014-07-22 22:43:51,807 INFO (ApplicationStartup.java:36) com.akbank.bsa.core.startup.SSLProviderInitializer - initializer end
2014-07-22 22:43:51,807 INFO (ApplicationStartup.java:34) com.akbank.bsa.core.startup.DatasourceDetector - initializer starting
2014-07-22 22:43:51,812 INFO (ApplicationStartup.java:36) com.akbank.bsa.core.startup.DatasourceDetector - initializer end
2014-07-22 22:43:51,812 INFO (ApplicationStartup.java:34) com.akbank.bsa.core.startup.ServiceRegistry - initializer starting
2014-07-22 22:43:52,502 INFO (ApplicationStartup.java:36) com.akbank.bsa.core.startup.ServiceRegistry - initializer end
2014-07-22 22:43:52,502 INFO (ApplicationStartup.java:34) com.akbank.bsa.core.startup.MqDetector - initializer starting
2014-07-22 22:43:55,900 INFO (ApplicationStartup.java:36) com.akbank.bsa.core.startup.MqDetector - initializer end
2014-07-22 22:43:55,901 INFO (ApplicationStartup.java:34) com.akbank.bsa.core.startup.SpringApplicationContext - initializer starting
2014-07-22 22:44:10,106 INFO (ApplicationStartup.java:36) com.akbank.bsa.core.startup.SpringApplicationContext - initializer end
2014-07-22 22:44:10,106 INFO (ApplicationStartup.java:34) com.akbank.bsa.core.refdata.ReferenceData - initializer starting
2014-07-22 22:44:10,132 INFO (ApplicationStartup.java:36) com.akbank.bsa.core.refdata.ReferenceData - initializer end
2014-07-22 22:44:10,133 INFO (ApplicationStartup.java:34) com.akbank.bsa.core.cache.CacheFactory - initializer starting
2014-07-22 22:44:10,171 INFO (ApplicationStartup.java:36) com.akbank.bsa.core.cache.CacheFactory - initializer end
2014-07-22 22:44:10,171 INFO (ApplicationStartup.java:34) com.akbank.bsa.core.scheduler.Scheduler - initializer starting
2014-07-22 22:44:10,178 INFO (ApplicationStartup.java:36) com.akbank.bsa.core.scheduler.Scheduler - initializer end
2014-07-22 22:44:10,179 INFO (ApplicationStartup.java:34) com.akbank.bsa.core.jms.MqListenerStarter - initializer starting
2014-07-22 22:44:10,236 INFO (ApplicationStartup.java:36) com.akbank.bsa.core.jms.MqListenerStarter - initializer end
2014-07-22 22:44:10,237 INFO (ApplicationStartup.java:34) com.akbank.infrastructure.batch.core.QuartzScheduler - initializer starting
2014-07-22 22:44:10,237 INFO (ApplicationStartup.java:36) com.akbank.infrastructure.batch.core.QuartzScheduler - initializer end
2014-07-22 22:44:46,252 INFO (ApplicationStartup.java:22) BSA Application started successfully.
```

WhitePlugin

Güncelleme

workspace.support projesi update alınır. C:\dev\eclipse\workspace\workspace.support\whiteplugin\plugins folder'ı altındaki son versiyon kopyalanarak C:\dev\eclipse\plugins altına kopyalanır. Eski versiyon WhitePlugin silinir. (Eclipse açıkken dosya kullanıldığı için silinemeyebilir, eclipse kapatıldıktan sonra silmek gerekiyor.)

BSA Server'larında çalışan uygulamalar

Component	Proje	Cluster	Datasource	Queue Connection Factory ve Queue
integro	MyNotes	Integro	jdbc/crmDS	
integro	MyWalletsStates	Integro	jdbc/crmDS	
integro	MyTargets	Integro	jdbc/crmDS	
integro	ScanRequirement	Integro	jdbc/bsa02DS	
integro	SysMonitoring	Integro	jdbc/sysMonitoringDS	
integro	AkrobatCommon	Integro	jdbc/subDS	
integro	Salesboard	Integro	jdbc/crmDS jdbc/bsa02DS	
integro	IntelligentDiary	Merkur-Batch	jdbc/bsa02DS	
integro	Profile	Integro	jdbc/bsa01DS	
integro	CreditActions	Integro	jdbc/bsa01DS	
integro	AuthorizationIntegrator	MKY	jdbc/subDS	
integro	BankReceipt	Merkur	jdbc/bsa02DS	
bpm	KBDS	Merkur-Batch	jdbc/bsa02DS	
corebanking	Accounting	Batch	jdbc/hostDS	
crm	Agenda	Merkur	jdbc/bsa02DS	
dys	CoreServices	Merkur	jdbc/dysDS jdbc/dysArchiveDS	jms/bsaDysQueueCF jms/bsaDysQueue
enterprise	Bps	Merkur	jdbc/bsa01DS	
enterprise	IKHost	Merkur	jdbc/hostDS	
enterprise	Mobile	Merkur-Batch	jdbc/mobilDS	
erp	ERPIk	ERP		
infrastructure	AdministrativeOperations	MKY	jdbc/hostDS	
infrastructure	Authorization	MKY	jdbc/bsa01DS	jms/bsaMkyLoginLogQueueCF jms/bsaMkyLoginLogRq
infrastructure	LdapAccessBase	MKY		
infrastructure	LdapWebService	MKY		
infrastructure	Batch	Batch	jdbc/bsa01DS	
infrastructure	HostMQLogger	Merkur	jdbc/bsaLogDS	jms/bsaHostQueueCF jms/hostMqLogQueue

integromobile	CommonOperations	Merkur-Batch	jdbc/mobilDS jdbc/hostDS jdbc/fraudDS	
investment	ForwardsSwap	Merkur	jdbc/bsa01DS	
investment\fx	FXCore	Investment	jdbc/bsa01DS jdbc/hostDS	
investment	WmpMeetingRecord	Merkur	jdbc/bsa02DS	
investment	WmpNotes	Merkur	jdbc/bsa02DS	
investment	WmpTasks	Merkur	jdbc/bsa02DS	
investment\portfolio	AccountCore	Investment	jdbc/akyGeneksDS	
investment\portfolio	EquityCore	Investment	jdbc/akbGeneksDS jdbc/hostDS jdbc/akbForeksDelayedDS jdbc/akbForeksDS	
loan\consumer	ProductDigitalization	Integro	jdbc/bsa01DS	
reftaskman	Financial	Merkur-Batch		
reftaskman	NonFinancial	Merkur-Batch	jdbc/mobilDS	
reftaskman	TaskManagerCore	Merkur-Batch	jdbc/bsa01DS	
workstation	Journal	Integro	jdbc/subDS	
dwh	Informatica	Merkur	jdbc/informaticaDS	
cards	CYS	Merkur-Batch	jdbc/bsa02DS	
cards\merchantrelations\ocean	CysGateway	Merkur-Batch	jdbc/oceanDS	
enterprise	MTS	Merkur	jdbc/bsa02DS	
branchapps	Sys	Merkur	jdbc/sysDS	
loan\consumer	CalculationTools	Merkur		
enterprise	Geocoding	Merkur	jdbc/geocodingDS	
customer	Analytics	Merkur	jdbc/customerAnalyticsDS	

MCA - BSA servis ilişki tablosu

\\Sn000001\sys\Data\Project\2012Proj\NewBackendArchitectureFramework\ApplicationServiceFramework\MCA_BSA.xlsx

"http://tfssrv01:8080/wiki/index.php/BSA_Kullan%C4%B1c%C4%B1_K%C4%B1lavuzu"dan alındı

- Bu sayfa son olarak 07:59, 9 Mart 2017 tarihinde güncellenmiştir.