All@Pay™ 정기과금

PC 연동 매뉴얼

(Version 1.0.0.0)

www.allatpay.com

KG 올앳

2019년 04월

목차

1. 정기 과금(Regular Billing) 서비스

- **신용카드**를 이용한 결제 서비스로 고객의 동의 하에 공인인증서를 이용한 카드 인증 후, 고객의 카드 정보를 이용하여 상점이 원하는 시점에 매월 정기적으로 결제하는 서비스입니다.
- 매월 정기적으로 납부하는 요금, 이용료, 회비 청구 등 반복적인 결제시에 편리합니다.

2. 지원 OS 및 용어정의

2.1 지원가능한 OS 및 Browser

- Browser 및 version 별로 제한은 없습니다.
- OS 는 Windows 사용 가능합니다.

2.2 용어 정의

- Before Page : 각 기능에 필요한 값을 설정하는 페이지
- After Page : 올앳서버와 통신 및 결과값을 수신하는 페이지
- Receive Page : 결제창으로부터 결제 인증 결과값을 수신하는 페이지
- Extra : Web Page 없이 승인 요청을 보내는 모듈

3. 설치안내

3.1 상점 상태 확인

- 올앳페이 Home > 관리자 시스템(로그인) > 마이페이지 > 상점정보 > 상세보기 > 회원사 운영정보

* 상점 ID 의 오픈 여부를 확인한다. (오픈 : 올앳페이와 통신가능)
* 결제 연동에 필요한 상점 ID 와 Cross Key 를 확인한다.

3.2 설치안내

1) * 샘플 소스로부터 승인에 필요한 파일들을 확인합니다.
JSP 를 예시로 들었지만 다른 언어도 가능합니다. (JSP, PHP, ASP)

* JSP

파일명	설명
nonre_fix.html	카드키 등록 페이지 (Before Page)
nonre_fix_cancel.html	등록된 카드키 해지 페이지 (Before Page)
allat_receive.jsp	상점이 인증결과를 받는 페이지 (Receive Page)
allat_fix.jsp	카드키 등록 요청을 보내는 페이지 (After Page)
allat_fix_cancel.jsp	등록된 카드키 해지 요청을 보내는 페이지 (After Page)
AllatApprovalExtra.java	등록된 카드키를 통해 결제 승인 요청 소스 (Extra)
AllatCancelExtra.java	결제 승인된 거래 취소 요청 소스 (Extra)

AllatFixCancelExtra.java	등록된 카드키 해지 요청 소스 (Extra)
AllatUtil.java	올앳과 Socket 통신하기 위한 모듈 (내용 수정 금지)

2) AllatUtil.java 파일을 상점측 쇼핑몰의 웹서버로 업로드 후, 파일 상단에 package 구문을 추가하고 컴파일(.class 생성) 하면 설치가 완료됩니다. 권장 경로는 Bean 클래스가 있는 위치입니다.

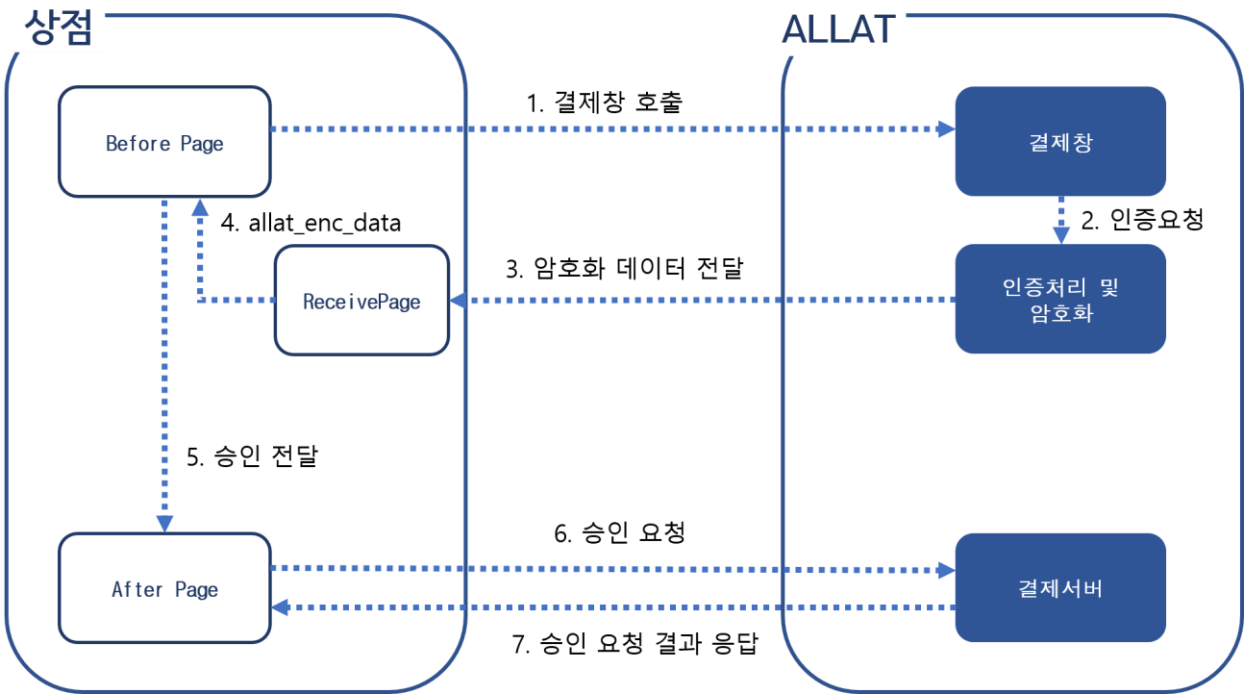
3) Before Page를 참조해 상점 측의 정기과금 카드키 등록 및 해지 페이지를 작성합니다.
웹 방식을 사용하지 않으신다면 Extra라는 이름이 명명된 소스로 연동하셔도 됩니다.
카드키 등록 페이지란 사용자로부터 정기과금 서비스를 제공하기 위하여 정기결제용 카드키를 발급받는 페이지입니다. 필요한 해당 필드를 설정하고 form 정보에 담아 진행하시면 됩니다.

4) Receive Page 파일(allat_receive.jsp)은 인증결과를 받는 페이지입니다.
Before Page의 result_submit 함수의 이름이 변경되지 않는다면 수정사항 없이 사용하시면 됩니다.
인증결과를 받는 페이지란 결제창을 통하여 인증된 결과를 받는 페이지입니다.
인증결과로 받는 항목으로 결과코드, 결과메세지, 암호화된 데이터가 있습니다.

5) After Page를 참조해 상점 측의 승인 요청 페이지를 작성합니다.
(Before Page에 따라 nonre_fix.html – allat_fix.jsp, nonre_fix_cancel.html – allat_fix_cancel.jsp)
승인 페이지란 인증완료 후 암호화된 정보들을 올앳 서버로 요청하여 승인을 받는 페이지입니다.
올앳 서버로 카드키 관련 요청을 보내기 위해서는 필수항목 3 가지가 있습니다.
allat_shop_id(상점 ID), allat_enc_data(암호화된 정보), allat_cross_key(상점이 가지고 있는 고유한 키값)
위 3 가지 항목을 소켓통신을 이용하여 올앳 서버로 전송하게 됩니다.

3.3 프로세스

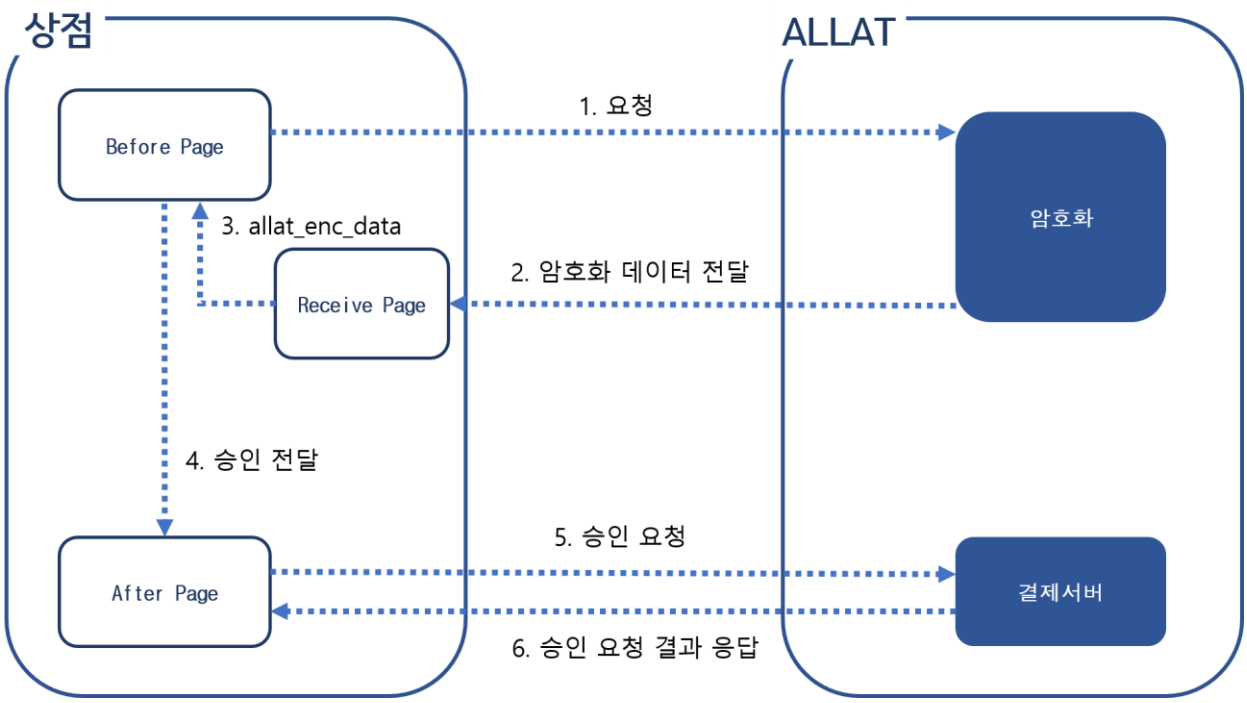
카드키 등록 요청



1) 올앳에서 제공한 JavaScript 함수(Allat_Plus_Fix(dfm, "0", "0"))를 호출하여 페이지에 설정한 값으로 올앳 카드키 등록창을 띄웁니다.

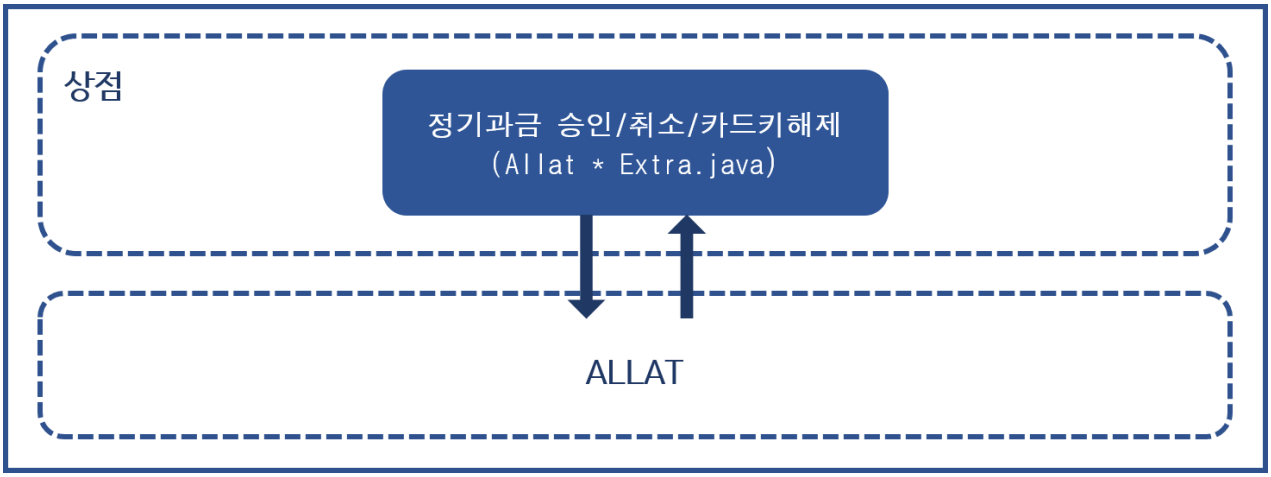
2) 카드키 등록창에서 카드키 등록 인증 과정을 진행합니다.
3) 등록 인증 과정이 완료되고 결과로 내려오는 암호화된 인증정보는 상점 측에서 구현한 Receive Page로 전달됩니다.
(Before Page에서 전달받은 shop_receive_url의 값을 URL로 사용, 접속 가능한 Full URL 입력)
4) Receive Page에서 JavaScript를 이용하여 Before Page의 함수를 호출, 암호화된 인증정보 전달합니다.
5) Before Page에서 전달받은 값을 allat_enc_data 필드에 세팅하여 After Page로 승인요청 값 전달합니다.
6) After Page에서 전달받은 값들을 이용하여 올앳 측의 결제서버로 카드 등록 승인 요청
7) 승인 요청 처리 후 결과 응답 값을 처리합니다.
* 6~7 과정은 올앳에서 제공한 AllatUtil 을 사용해서 소켓통신이 이루어지게 됩니다.

카드키 해지 요청



1) 올앳에서 제공한 JavaScript 함수(Allat_Plus_Fix_Cancel(dfm, "0", "0"))를 호출하여 페이지에 설정한 값으로 카드키 해지 요청을 합니다.
2) 페이지에 설정된 값을 암호화하여 상점 측에서 구현한 Receive Page로 전달됩니다.
(Before Page에서 전달받은 shop_receive_url의 값을 URL로 사용, 접속 가능한 Full URL 입력)
3) Receive Page에서 JavaScript를 이용하여 Before Page의 함수를 호출, 암호화된 인증정보 전달합니다.
4) Before Page에서 전달받은 값을 allat_enc_data 필드에 세팅하여 After Page로 승인요청 값 전달합니다.
5) After Page에서 전달받은 값들을 이용하여 올앳 측의 결제서버로 카드키 해지 승인 요청
6) 취소 승인 요청 처리 후 결과 응답 값을 상점 측의 After Page로 전송
* 5~6 과정은 올앳에서 제공한 AllatUtil 을 사용해서 소켓통신이 이루어지게 됩니다.

정기과금 거래건 승인, 취소 및 카드키 해제 요청 (Extra)



1) 소스에 있는 결제 요청 정보 필드를 작성합니다.
2) 작성한 뒤 컴파일하여 실행합니다.
3) AllatUtil 모듈을 통해 작성한 정보를 가지고 승인/취소/카드키 해제가 이루어지게 됩니다.
4) 실행 후 받아오는 응답 값을 처리합니다.

3.4 테스트

1) AfterPage(카드키 등록은 allat_fix.jsp, 카드키 해제(allat_fix_cancel.jsp) 소스내에 필요한 필드 값 입력)
2) 웹 브라우저를 이용하여 서버의 nonre_fix.html, nonre_fix_cancel.html 호출
3) 필수정보 입력
4) 옵션정보의 테스트 여부 값을 'Y'로 설정(올앳과 통신 확인용, 실제 승인은 이루어지지 않습니다.)
5) 인증 버튼을 클릭하여 인증 진행
6) 결과코드 '0001', 결과메세지 '테스트성공' 확인(실제 승인은 '0000'이 정상코드입니다.)

4. 정기과금 소스 설명

4.1 nonre_fix.html (Before Page)

```
<html>
<head>
<meta http-equiv="content-type" content="text/html; charset=euc-kr">
<script language=JavaScript charset='euc-kr' src="http://tx.allatpay.com/common/NonAllatPayREPlus.js"></script>
<script language=Javascript>
// 인증 요청
function ftn_fix(dfm) {
        Allat_Plus_Fix(dfm, "0", "0");
}
// 결과값 설정
function result_submit(result_cd,result_msg,enc_data) {
        Allat_Plus_Close();
        // 정상적인 인증이 되지 않았을 때
        if(result_cd != '0000') {
        alert(result_cd + " : " + result_msg);
        // 정상적인 인증일 때
        } else {
```

```
                        fm.allat_enc_data.value = enc_data;
                        fm.action = "allat_fix.jsp";
                        fm.method = "post";
                        fm.target = "_self";
                        fm.submit();
                }
}
</script>

</head>
<body>
    <form name="fm"    method=POST action="allat_fix.jsp">
    // 필수 정보
    <input type=text name="allat_shop_id" value="" size="19" maxlength=20>          // 상점 ID
    <input type=text name="allat_order_no" value="" size="19" maxlength=70>         // 주문번호
    <input type="text" name="shop_receive_url" value="" size="19">                  // Receive Page URL
    <input type=hidden name=allat_enc_data value=''>                                // 인증정보 저장할 필드
    // 옵션 정보
    <input type=text name="allat_pmember_id" value="" size="19" maxlength=20>      // 회원 ID
    <input type=text name="allat_amt" value="" size="19" maxlength=20>              // 금액
    <input type=text name="allat_product_nm" value="" size="19" maxlength=100>     // 제품명
    <input type=text name="allat_email_addr" value="" size="19" maxlength=50>      // 이메일 주소
    <input type=text name="allat_registry_no" value="" size="19" maxlength=13>     // 주민번호
    <input type=text name="allat_fix_type" value="" size="19" maxlength=3>         // 정기과금 타입(FIX/HOF)
    <input type=text name="allat_test_yn" value="N" size="19" maxlength=1>         // 테스트 여부(Y/N)

    <input type=button value="인증요청" name=app_btn onClick="javascript:ftn_fix(document.fm);">
</form>
</body>
</html>
```

4.2 nonre_fix_cancel.html (Before Page)

```
<html>
<head>
<meta http-equiv="content-type" content="text/html; charset=euc-kr">
<script language=JavaScript charset='euc-kr' src="http://tx.allatpay.com/common/NonAllatPayREPlus.js"></script>
<script language=Javascript>
function ftn_fix_cancel(dfm) {
        Allat_Plus_Fix_Cancel(dfm, "0", "0");
}
function result_submit(result_cd,result_msg,enc_data) {
        Allat_Plus_Close();
        if(result_cd != '0000') {
                alert(result_cd + " : " + result_msg);
        } else {
                fm.allat_enc_data.value = enc_data;
                fm.action = "allat_fix_cancel.jsp";
                fm.method = "post";
                fm.target = "_self";
        fm.submit();
        }
}
</script>
</head>

<body>
```

```
    <form name="fm"    method=POST action="allat_fix_cancel.jsp">
    <table border=0 cellpadding=0 cellspacing=1 bgcolor="#606060" width=90% align=center>
    // 필수 정보
    <input type=text name="allat_shop_id" value="" size="19" maxlength=20>          // 상점 ID
    <input type=text name="allat_fix_key" value="" size=26 maxlength=24></td>       // 해지할 카드키
    <input type="text" name="shop_receive_url" value="" size="19">                  // Receive Page URL
    <input type=hidden name=allat_enc_data value=''>                                // 인증정보 저장할 필드
    <input type=hidden name=allat_opt_pin value="NOVIEW">                           // 올앳 참조 필드(수정없이 사용)
    <input type=hidden name=allat_opt_mod value="WEB">                              // 올앳 참조 필드(수정없이 사용)

    // 옵션 정보
    <input type=text name="allat_fix_type" value="" size=26 maxlength=3>            // 정기과금 타입(FIX/HOF)
    <input type=text name="allat_test_yn" value="N" size=26 maxlength=1>            // 테스트 여부(Y/N)

    <input type=button value="카드키 해지 요청" name=app_btn onClick="javascrt:ftn_fix_cancel(document.fm);">
    </form>
</body>
</html>
```

4.3 allat_receive.jsp (Receive Page)

* 결제가 끝나고 결제창에서 submit 되는 페이지이므로 FullURL 입력 필수

```
<%@ page contentType="text/html; charset=euc-kr" %>
<%
// 결과값 설정
String sResultCd    = request.getParameter("allat_result_cd");
String sResultMsg = request.getParameter("allat_result_msg");
String sEncData     = request.getParameter("allat_enc_data");

// 설정한 결과값을 Before Page 의 result_submit 함수 호출하며 결과값 전달
out.println("<script>");
out.println("if(window.opener != undefined) {");
out.println("            opener.result_submit('" + sResultCd + "','" + sResultMsg + "','" + sEncData + "');");
out.println("            window.close();");
out.println("} else {");
out.println("            parent.result_submit('" + sResultCd + "','" + sResultMsg + "','" + sEncData + "');");
out.println("}");
out.println("</script>");
%>
```

4.4 allat_fix.jsp (After Page)

```
<%-- AllatUtil Import, 상점에 설치한 경로에 맞게 import 하셔야 합니다 --%>
<%@ page import="java.util.*,java.net.*,com.allat.util.AllatUtil" %>
<%
  // 필수 설정 값
  String sCrossKey   = "상점 크로스키";
  String sShopId     = "상점 아이디";
  //Before Page 로부터 전달받은 인증결과 값 설정
  String sEncData    = request.getParameter("allat_enc_data");
  String strReq = "";

  // 요청 데이터 설정
  strReq    ="allat_shop_id="+sShopId;
  strReq +="&allat_enc_data="+sEncData;
  strReq +="&allat_cross_key="+sCrossKey;

  // 올앳 결제서버와 통신 : AllatUtil.CertRegReq-> 카드키 등록 통신함수, HashMap->결과값
  AllatUtil util = new AllatUtil();
  HashMap hm = null;
  hm = util.CertRegReq(strReq, "SSL"); // SSL(443 포트) 통신 불가 시 "SSL"을 "NOSSL"로 변경
                                       NOSSL 로 변경 시 80 포트 이용

  // 요청 결과 값
  String sReplyCd        = (String)hm.get("reply_cd");
  String sReplyMsg       = (String)hm.get("reply_msg");

  /* 결과값 처리
  ----------------------------------------------------------------------
     결과 값(sReplyCd)이 '0000'이면 정상. 단, BeforePage 의 allat_test_yn=Y 일경우 '0001'이 정상.
     실제 승인    : allat_test_yn=N 일 경우 sReplyCd=0000 이면 정상
     테스트 승인 : allat_test_yn=Y 일 경우 sReplyCd=0001 이면 정상
  ----------------------------------------------------------------------*/
  if( sReplyCd.equals("0000") ){
    // sReplyCd "0000" 일 때만 성공
    String sFixKey             = (String)hm.get("fix_key");
    String sApplyYmd           = (String)hm.get("apply_ymd");
    String sCardNo             = (String)hm.get("card_no");
    // 이하 생략, 상점에서 결제 성공일 때의 화면 처리
  }else{
    // sReplyCd 가 "0000" 아닐 때는 에러
    // sReplyMsg 결과에 대한 메세지
    out.println("결과코드 : " + sReplyCd    + "<br>");
    out.println("결과메세지 : " + sReplyMsg + "<br>");
  }
%>
```

4.5 allat_fix_cancel.jsp (After Page)

```
<%-- AllatUtil Import, 상점에 설치한 경로에 맞게 import 하셔야 합니다 --%>
<%@ page import="java.util.*,java.net.*,com.allat.util.AllatUtil" %>
<%
  // 필수 설정 값
  String sCrossKey   = "상점 크로스키";
  String sShopId     = "상점 아이디";
  //Before Page 로부터 전달받은 인증결과 값 설정
  String sEncData    = request.getParameter("allat_enc_data");
  String strReq = "";

  // 요청 데이터 설정
  strReq    ="allat_shop_id="+sShopId;
  strReq +="&allat_enc_data="+sEncData;
  strReq +="&allat_cross_key="+sCrossKey;

  // 올앳 결제서버와 통신 : AllatUtil.CertRegReq-> 카드키 등록 통신함수, HashMap->결과값
  AllatUtil util = new AllatUtil();
  HashMap hm = null;
  hm = util.CertCancelReq(strReq, "SSL"); // SSL(443 포트) 통신 불가 시 "SSL"을 "NOSSL"로 변경
                                          NOSSL 로 변경 시 80 포트 이용

  // 결제 결과 값
  String sReplyCd        = (String)hm.get("reply_cd");
  String sReplyMsg       = (String)hm.get("reply_msg");

  /* 결과값 처리
  ----------------------------------------------------------------------
     결과 값(sReplyCd)이 '0000'이면 정상. 단, BeforePage 의 allat_test_yn=Y 일경우 '0001'이 정상.
     실제 승인    : allat_test_yn=N 일 경우 sReplyCd=0000 이면 정상
     테스트 승인 : allat_test_yn=Y 일 경우 sReplyCd=0001 이면 정상
  ----------------------------------------------------------------------*/
  if( sReplyCd.equals("0000") ){
    // sReplyCd "0000" 일 때만 성공
    String sFixKey             = (String)hm.get("fix_key");
    String sApplyYmd           = (String)hm.get("apply_ymd");
    // 이하 생략, 상점에서 결제 성공일 때의 화면 처리
  }else{
    // sReplyCd 가 "0000" 아닐 때는 에러
    // sReplyMsg 결과에 대한 메세지
    out.println("결과코드 : " + sReplyCd    + "<br>");
    out.println("결과메세지 : " + sReplyMsg + "<br>");
  }
%>
```

4.6 AllatApprovalExtra.java (Extra)

```
// AllatUtil Import, 상점에 설치한 경로에 맞게 import 하셔야 합니다
import java.util.*;
import java.io.*;
import com.allat.util.*;

public class AllatApprovalExtra {

  ///Fields


  ///Constructor
  public AllatApprovalExtra(){};
  ///Method


  ///Main Method

  public static void main(String args[]){

    HashMap reqHm=new HashMap();
    HashMap resHm=null;
    String szReqMsg="";
    String szAllatEncData="";
    String szCrossKey="";

    // 예제 소스를 참고(매뉴얼에선 내용 생략)
    // 결제 요청 정보 필드 선언
    // 결제 요청 정보 값 설정
    // 결제 요청 정보 HashMap 에 put

    AllatUtil util = new AllatUtil();

    // 결제 요청 정보 암호화
    szAllatEncData=util.setValue(reqHm);

    szReqMsg    = "allat_shop_id="      + szShopId
                + "&allat_amt="          + szAmt
                + "&allat_enc_data=" + szAllatEncData
                + "&allat_cross_key="+ szCrossKey;

    // 올앳 결제서버와 통신 : AllatUtil.approvalReq-> 결제 승인 요청 통신함수, HashMap->결과값
    resHm = util.approvalReq(szReqMsg, "SSL");

    String sReplyCd      = (String)resHm.get("reply_cd");
    String sReplyMsg    = (String)resHm.get("reply_msg");

    // reply_cd "0000" 일때만 성공
    if( sReplyCd.equals("0000") ){
      // 응답값 String 으로 get, 결과 처리 (예제 샘플 참고)
    }else{
      // reply_cd 가 "0000" 아닐때는 에러
      // reply_msg 가 실패에 대한 메세지
      System.out.println("결과코드            : " + sReplyCd     );
      System.out.println("결과메세지          : " + sReplyMsg    );
    }
  }
}
```

4.7 AllatCancelExtra.java (Extra)

```
// AllatUtil Import, 상점에 설치한 경로에 맞게 import 하셔야 합니다
import java.util.*;
import java.io.*;
import com.allat.util.*;

public class AllatCancelExtra {

  ///Fields


  ///Constructor
  public AllatCancelExtra(){};
  ///Method


  ///Main Method

  public static void main(String args[]){

    HashMap reqHm=new HashMap();
    HashMap resHm=null;
    String szReqMsg="";
    String szAllatEncData="";
    String szCrossKey="";

    // 예제 소스를 참고(매뉴얼에선 내용 생략)
    // 결제 요청 정보 필드 선언
    // 결제 요청 정보 값 설정
    // 결제 요청 정보 HashMap 에 put

    AllatUtil util = new AllatUtil();

    // 결제 요청 정보 암호화
    szAllatEncData=util.setValue(reqHm);

    szReqMsg    = "allat_shop_id="      + szShopId
                + "&allat_amt="         + szAmt
                + "&allat_enc_data=" + szAllatEncData
                + "&allat_cross_key="+ szCrossKey;

    // 올앳 결제서버와 통신 : AllatUtil.cancelReq-> 결제 취소 요청 통신함수, HashMap->결과값
    resHm = util.cancelReq(szReqMsg, "SSL");

    String sReplyCd     = (String)resHm.get("reply_cd");
    String sReplyMsg    = (String)resHm.get("reply_msg");

    // reply_cd "0000" 일때만 성공
    if( sReplyCd.equals("0000") ){
      // 응답값 String 으로 get, 결과 처리 (예제 샘플 참고)
    }else{
      // reply_cd 가 "0000" 아닐때는 에러
      // reply_msg 가 실패에 대한 메세지
      System.out.println("결과코드            : " + sReplyCd     );
      System.out.println("결과메세지          : " + sReplyMsg    );
    }
  }
}
```

4.8 AllatFixCancelExtra.java (Extra)

```
// AllatUtil Import, 상점에 설치한 경로에 맞게 import 하셔야 합니다
import java.util.*;
import java.io.*;
import com.allat.util.*;

public class AllatFixCancelExtra {

   ///Fields


   ///Constructor
   public AllatFixCancelExtra(){};
   ///Method


   ///Main Method

   public static void main(String args[]){

      HashMap reqHm=new HashMap();
      HashMap resHm=null;
      String szReqMsg="";
      String szAllatEncData="";
      String szCrossKey="";

      // 예제 소스를 참고(매뉴얼에선 내용 생략)
      // 결제 요청 정보 필드 선언
      // 결제 요청 정보 값 설정
      // 결제 요청 정보 HashMap 에 put

      AllatUtil util = new AllatUtil();

      // 결제 요청 정보 암호화
      szAllatEncData=util.setValue(reqHm);

      szReqMsg    = "allat_shop_id="      + szShopId
                  + "&allat_enc_data=" + szAllatEncData
                  + "&allat_cross_key="+ szCrossKey;

      // 올앳 결제서버와 통신 : AllatUtil.CertCancelReq-> 결제 취소 요청 통신함수, HashMap->결과값
      resHm = util.CertCancelReq(szReqMsg, "SSL");

      String sReplyCd     = (String)resHm.get("reply_cd");
      String sReplyMsg    = (String)resHm.get("reply_msg");

      // reply_cd "0000" 일때만 성공
      if( sReplyCd.equals("0000") ){
        // 응답값 String 으로 get, 결과 처리 (예제 샘플 참고)
      }else{
        // reply_cd 가 "0000" 아닐때는 에러
        // reply_msg 가 실패에 대한 메세지
        System.out.println("결과코드            : " + sReplyCd     );
        System.out.println("결과메세지          : " + sReplyMsg    );
      }
   }
}
```

5. 주의사항

1) BeforePage 의 NonAllatPayREPlus.js 파일은 다운받아서 사용하지 마십시오.
(지속적으로 업데이트 되는 소스입니다)
2) BeforePage 의 NonAllatPayREPlus.js 는 EUC-KR 로 읽어서 상점의 Encoding Type 에 맞게 Include 됩니다.
3) BeforePage 의 승인 버튼을 input type='submit', input type='image'를 사용하시면 안됩니다.
Submit 기능이 있어서 오류가 발생할 수 있습니다.
※ 이미지 사용시 "<a>" 태그 사용
예) <img src='이미지 경로'>

4) AfterPage 의 올앳과 통신하는 부분인 CertRegReq(), CertCancelReq() 등 해당 부분 호출 전/후로 로그를 남기는 것을 권장합니다.
5) 결과값을 DB 에 저장시 AfterPage 에서 처리할 것을 권장합니다.
다른 페이지로 이동하여 DB 처리를 하는 경우, 결과값이 유실되는 경우가 발생할 수 있습니다.
6) 상점이 호스팅정기과금으로 신청했을 시 카드키 등록은 호스팅 ID, 결제는 하위 상점 ID 로 하셔야합니다.
7) 상점이 일반 정기과금으로 신청했을 시 카드키 등록은 신청한 상점 ID, 결제도 동일한 상점 ID 로 하셔야 합니다.