IGAWorks 리워드서버 샘플코드

IGAWorks 리워드 서버에서 전달하는 리워드지급정보를 수신하여 처리한 후 처리 결과를 Json 타입으로 회신하여야 합니다.

<상황 별 Json 응답 포맷>

- 보안성검증실패 : {"Result":false,"ResultCode":1100,"ResultMsg":"invalid signed value"}
- 리워드중복지급 : {"Result":false,"ResultCode":3100,"ResultMsg":"duplicate transaction"}
- 유저검증실패 : {"Result":false,"ResultCode":3200,"ResultMsg":"invalid user "}
- 예외사항발생 : {"Result":false,"ResultCode":4000,"ResultMsg":"custom error message"}

- 리워드지급성공 : {"Result":true,"ResultCode":1,"ResultMsg":"success"}

<샘플 코드 제공>

C#, PHP 샘플 코드를 제공해드리며, 제공되지 않은 다른 개발 언어의 경우에는 위의 처리결과 포맷에 맞추어 처리 결과를 회신 해 주시면 됩니다.

C# sample code

```
// usn : 리워드를 지급할 유저 ID
// reward_key : 리워드 요청에 대한 transaction_id(각 리워드 요청당 unique)
// quantity : 리워드 지급량
// campaign_key : 참여 완료한 캠페인 키
// signed_value : 리워드 요청 보안 체크 값

string hash_key = "IGAWorks 에서 발급한 해시키 입력";

// < signed_value 체크 성공 >
if( signed_value == GetHMACMD5Hash(usn + reward_key + quantity + campaign_key, hash_key) )
{
      // 매체사 서버에서 IGAWorks 의 리워드 지급 요청을 처리하였음에도,
       // 네트워크 오류 등으로 인해 IGAWorks 서버에서 리워드 지급 요청이 실패했다고 판단하고 동일한 지급 요청을 다시 보내는 경우가 발생할 수 있음.
       // 이 때 매체사 서버에서는 IGAWorks 서버가 보낸 reward_key 가 이미 처리된 리워드 요청에 대한 reward_key 일 경우 아래와 같이 처리.
```

```
// < reward_key 에 해당하는 리워드 지급이 이미 완료 되었는지 체크 >
      if( // 이미 리워드 지급이 완료된 reward_key 일 경우 )
      {
          // < 리워드 중복 지급에 해당하는 Json string return >
           // {"Result":false,"ResultCode":3100,"ResultMsg":"duplicate transaction"}
      }
      else
      {
           // < 유저에게 리워드 지급 후 성공 Json string return >
          // {"Result":true,"ResultCode":1,"ResultMsg":"success"}
      }
}
// < signed_value 체크 실패 >
else
{
      // < 보안 체크 실패에 해당하는 Json string return >
       // {"Result":false,"ResultCode":1100,"ResultMsg":"invalid hash key"}
}
```

```
private string GetHMACMD5Hash(string plainText, string secretKey)
{
      // signed value check
==========================================================================
      byte[] secretkeyBytes = Encoding.ASCII.GetBytes(secretKey);
      byte[] plainTextBytes = Encoding.ASCII.GetBytes(plainText);
      HMACMD5 md5 = new HMACMD5(secretkeyBytes);
      // Compute the hash of the supplied query string:
      byte[] hashValue = md5.ComputeHash(plainTextBytes);

      // Convert the hash into a hexadecimal string for comparison:
      StringBuilder hashedString = new StringBuilder();
      for (int i = 0; i < hashValue.Length; i++)
      {
          hashedString.Append(hashValue[i].ToString("x2")); // "x2" means lowercase hexadecimal
      }

      return hashedString.ToString();
}
```

php sample code

```php
// $usn : 리워드를 지급할 유저 ID
// $reward_key : 리워드 요청에 대한 transaction_id(각 리워드 요청당 unique)
// $quantity : 리워드 지급량
// $campaign_key : 참여 완료한 캠페인 키
// $signed_value : 리워드 요청 보안 체크 값


$hash_key = <IGAWorks 에서 발급한 해시키 입력>;

// < signed_value 체크 성공 >
if( $signed_value == hash_hmac('md5', $usn . $reward_key . $quantity . $campaign_key, $hash_key))
{
    // 매체사 서버에서 IGAWorks 의 리워드 지급 요청을 처리하였음에도,
     // 네트워크 오류 등으로 인해 IGAWorks 서버에서 리워드 지급 요청이 실패했다고 판단하고 동일한 지급 요청을 다시 보내는 경우가 발생할 수 있음.
     // 이 때 매체사 서버에서는 IGAWorks 서버가 보낸 reward_key 가 이미 지급 처리된 리워드 요청에 대한 reward_key 일 경우 아래와 같이 처리.

    // < reward_key 에 해당하는 리워드 지급이 이미 완료 되었는지 체크 >
    if( // 이미 리워드 지급이 완료된 reward_key 일 경우 )
    {
        // < 리워드 중복 지급에 해당하는 Json string return >
         // {"Result":false,"ResultCode":3100,"ResultMsg":"duplicate transaction"}
    }  else   {
        // < 유저에게 리워드 지급 후 성공 Json string return >
         // {"Result":true,"ResultCode":1,"ResultMsg":"success"}
    }
} else   {
        // < signed_value 체크 실패 >
    // < 보안 체크 실패에 해당하는 Json string return >
     // {"Result":false,"ResultCode":1100,"ResultMsg":"invalid hash key"}
}
```