**[SWORLD]**

**Client Method Specification**

|  |  |
| --- | --- |
| **Ngày tạo** | [26/06/2013] |
| **Phiên bản** | [1.0] |
| **Tình trạng** | [Đề nghị] |
| **Tác giả** | [Tiêu Hải Minh, Developer] |
| **Duyệt bởi** | [] |
| **Ban hành bởi** | [] |

# LỊCH SỬ PHIÊN BẢN

| **Ngày** | **Phiên bản** | **Mô tả** | **Tác giả** |
| --- | --- | --- | --- |
|  |  | | |
| 26/06/2013 | 1.0 | Khởi tạo tài liệu | Tiêu Hải Minh |
|  |  |  |  |
|  |  |  |  |

# NỘI DUNG

[LỊCH SỬ PHIÊN BẢN 2](#_Toc360022079)

[NỘI DUNG 3](#_Toc360022080)

[1 Nhập thẻ 4](#_Toc360022081)

[2 Cấp thẻ 4](#_Toc360022082)

[3 Đọc dữ liệu thẻ 5](#_Toc360022083)

[4 Xóa dữ liệu thẻ 5](#_Toc360022084)

[5 Thống kê thẻ 6](#_Toc360022085)

[6 Danh sách thẻ 6](#_Toc360022086)

[7 Danh sách lượt phát hành 7](#_Toc360022087)

# Nhập thẻ

* Những phương thức client gọi đến server:
* DataForImportCard CheckAndGetDataToImportCard(string sessionId, byte[] serialNumber, int cardType)
* Nhiệm vụ: Kiểm tra thẻ tồn tại trong hệ thống và lấy dữ liệu từ database trả về cho client.
* Module hiện thực phương thức: CMS
* Input:
* sessionId: session của user
* Serialnumber: số serial của thẻ
* Cardtype: loại thẻ
* Output:
* class DataForImportCard
* CardDto ImportCard(string sessionId, byte[] serialNumber, int cardType, byte hmkAlias, byte dmkAlias)
* Nhiệm vụ: Ghi dữ liệu của thẻ vào database và trả đối tượng CardDto về cho client
* Module hiện thực phương thức: CMS
* Input:
* sessionId: session của user
* cardType: Loại thẻ
* hmkAlias: trường đinh danh của Header Master Key
* dmkAlias: trường định dan của data master key
* Output:
* Class CardDto

# Cấp thẻ

* Những phương thức client gọi đến server:
* void PersoCard(string sessionId, long MemberId, string serialNumberHex)
* Nhiệm vu: cá thể hóa thẻ cho thành viên, và ghi thông tin ứng dụng vào sector của thẻ
* Module hiện thực phương thức: PS
* Input:
* sessionId: session của user
* MemberId: Mã thành viên
* serialNumberHex: mã hex số serial number của thẻ (chuyển từ dạng int sang dạng hex)
* Output: không.
* DataForPersoCard CheckAndGetDataToPersoCard(string sessionId, long MemberId, byte[] serialNumber, byte hmkAlias, byte dmkAlias)
* Nhiệm vụ: Kiểm tra thẻ tồn tại trong hệ thống và lấy dữ liệu từ database trả về cho client
* Module hiện thực phương thức: PS
* Input:
* sessionId: sessionId của user
* memberId: Mã thành viên
* serialNumber: Mã serial của thẻ
* hmkAlias: trường đinh danh của Header Master Key
* dmkAlias: trường định dan của data master key
* Output:
* Class DataForPersoCard

# Đọc dữ liệu thẻ

* Những phương thức client gọi đến server:
* DataForReadCard GetDataToReadCard(string sessionId, byte[] serialNumber, int cardType, List<AppMetadataDto> appMetadataList, byte curDmkAlias)
* Nhiệm vụ: lấy dữ liệu từ database dựa trên serialNumber của thẻ và trả kết quả về cho client.
* Module hiện thực phương thức: CMS
* Input:
* sessionId: mã session của user
* serialNumber: mã serial của thẻ
* cardType: loại thẻ
* appMetadatalist: danh sách những ứng dụng đã ghi vào thẻ.
* curDmkAlias: data master key được ghi của thẻ
* Output:
* Class DataForReadCard
* StringCollection ParseTeacherAppData(string sessionId, byte[] MemberAppDataBytes)
* Nhiệm vụ: lấy dữ liệu của thành viên từ database trả về cho client
* Module hiện thực phương thức: PS
* Input:
* sessionId: session của user
* MemberAppDataByte: thông tin AppData.
* Output:
* Class StringCollection

# Xóa dữ liệu thẻ

* Những phương thức client gọi đến server:
* DataForClearCard CheckAndGetDataToClearCard(string sessionId, byte[] serialNumber, int cardType, byte curHmkAlias, byte curDmkAlias, List<AppMetadataDto> curAppList)
* Nhiệm vụ: kiểm tra và lấy dữ liệu của thẻ trả về cho client
* Module hiện thực phương thức: CMS
* Input:
* sessionId: session của user
* serialNumber: mã serial của thẻ
* cardType: loại thẻ
* curDmkAlias: data master key được ghi của thẻ
* curHmkAlias: Header master key được ghi của thẻ
* CurAppList: danh sách những ứng dụng đã ghi vào thẻ.
* Output:
* Class DataForClearCard
* void ClearCardData(string sessionId, byte[] serialNumber, byte hmkAlias, byte dmkAlias)
* Nhiệm vụ: xóa dữ liệu thẻ trên database
* Module hiện thực phương thức: CMS
* Input:
* sessionId: session của user
* serialNumber: mã serial của thẻ
* hmkAlias: trường đinh danh của Header Master Key
* dmkAlias: trường định dan của data master key
* Output: Không

# Thống kê thẻ

* Những phương thức client gọi đến server:
* List<CardStatisticsData> StatisticCardByPhysicalStatus(string sessionId)
* Nhiệm vụ: Thống kê danh sách thẻ theo trạng thái vật lý (bị hư, bị mất), trả kết quả về cho client
* Module hiện thực phương thức: CMS
* Input:
* sessionId: session của user
* Output:
* Trả về danh sách List
* List<CardStatisticsData> StatisticCardByPersoStatus(string sessionId)
* Nhiệm vụ: Thống kê danh sách thẻ theo trạng thái luận lý (thẻ chưa phát hành, thẻ đã phát hành)
* Module hiện thực phương thức: CMS
* Input:
* sessionId: session của user
* Output:
* Trả về danh sách List

# Danh sách thẻ

* Những phương thức client gọi đến server:
* List<CardDto> GetCardList(string sessionId, CardFilterDto filter, int skip, int take, out int totalRecords)
* Nhiệm vụ: Lấy danh sách thẻ đã được ghi vào hệ thống theo điều kiện filter
* Module hiện thực phương thức: CMS
* Input:
* sessionId: session của user
* filter: điều kiện lọc
* skip: vị trí record bắt đầu lấy dữ liệu
* take: số record lấy
* totalRecord: tổng số record lấy được
* Output:
* Trả về danh sách List

# Danh sách lượt phát hành

* Những phương thức client gọi đến server:
* List<PersonalizationDto> GetPersoList(string sessionId, PersoFilterDto filter, int skip, int take, out int totalRecords)
* Nhiệm vụ: lấy danh sách lượt thẻ đã phát hành
* Module hiện thực phương thức: PS
* Input:
* sessionId: session của user
* filter: điều kiện lọc
* skip: vị trí record bắt đầu lấy dữ liệu
* take: số record lấy
* totalRecord: tổng số record lấy được
* Output:
* Trả về danh sách List
* List<MethodResultDto> LockPersoes(string sessionId, long[] persoIds, string lockReason)
* Nhiệm vụ: khóa lượt phát hành thẻ
* Module hiện thực phương thức: PS
* Input:
* sessionId: session của user
* persoIds: mảng chứa danh sách những Id thẻ cần khóa
* lockReason: lý do khóa thẻ
* Ouput:
* Trả về danh sách List
* List<MethodResultDto> UnLockPersoes(string sessionId, long[] persoIds, string unlockReason)
* Nhiệm vụ: mở khóa lượt phát hành thẻ
* Module hiện thực phương thức: PS
* Input:
* sessionId: session của user
* persoIds: mảng chứa danh sách những Id thẻ cần khóa
* unlockReason: lý do mở khóa thẻ
* Ouput:
* Trả về danh sách List
* List<MethodResultDto> CancelPersoes(string sessionId, long[] persoIds, string cancelReason)
* Nhiệm vụ: hủy lượt phát hành thẻ
* Module hiện thực phương thức: PS
* Input:
* sessionId: session của user
* persoIds: mảng chứa danh sách những Id thẻ cần khóa
* cancelReason: lý do mở khóa thẻ
* Ouput:
* Trả về danh sách List
* List<MethodResultDto> ExtendPerso(string sessionId, long[] persoIds, DateTime expirationDate)
* Nhiệm vụ: gia hạn lượt phát hành thẻ
* Module hiện thực phương thức: PS
* Input:
* sessionId: session của user
* persoIds: mảng chứa danh sách những Id thẻ cần khóa
* expirationDate: Ngày hết hạn
* Output:
* Trả về danh sách List
* List<MethodResultDto> MarkBrokenCards(string sessionId, long[] cardIds)
* Nhiệm vụ: đánh đấu thẻ bị hư
* Module hiện thực phương thức: PS
* Input:
* sessionId: session của user
* cardIds: mảng chứa danh sách những Id thẻ cần đánh dấu
* Output:
* Trả về danh sách List
* List<MethodResultDto> UnMarkBrokenCards(string sessionId, long[] cardIds)
* Nhiệm vụ: hủy đánh đấu thẻ bị hư
* Module hiện thực phương thức: PS
* Input:
* sessionId: session của user
* cardIds: mảng chứa danh sách những Id thẻ cần hủy đánh dấu
* Output:
* Trả về danh sách List
* List<MethodResultDto> MarkLostCards(string sessionId, long[] cardIds);
* Nhiệm vụ: đánh đấu thẻ bị mất
* Module hiện thực phương thức: PS
* Input:
* sessionId: session của user
* cardIds: mảng chứa danh sách những Id thẻ cần đánh dấu
* Output:
* Trả về danh sách List
* List<MethodResultDto> UnMarkLostCards(string sessionId, long[] cardIds);
* Nhiệm vụ: hủy đánh đấu thẻ bị mất
* Module hiện thực phương thức: PS
* Input:
* sessionId: session của user
* cardIds: mảng chứa danh sách những Id thẻ cần hủy đánh dấu
* Output:
* Trả về danh sách List