

B+R-Buchungs- und Zugangssystem (NRW) der go.Rheinland GmbH und des Nahverkehr Westfalen-Lippe (NWL)

Beschreibung der Schnittstelle Hintergrundsystem Viaboxx und Terminal Version 1.1.3

Vorgelegt von:
Viaboxx GmbH
Hauptstr. 164
53639 Königswinter

Vertreten durch Jürgen Lüders, Geschäftsführer Viaboxx

im Auftrag der go.Rheinland GmbH und des Nahverkehr Westfalen-Lippe (NWL)



Dokumentenhistorie

Datum	Version*	Änderung	Autor
18.07.2023	1.1011	Formale Korrekturen	Jürgen Lüders
31.07.2023	1.102	Prüfung der Funktion	Rene Jeschke
31.07.2023	1.103	Review	Björn Golombek
31.07.2023	1.014	Merge	Jürgen Lüders
15.08.2023	1.015	Zusammenfassung aller Feedback	Jürgen Lüders
03.11.2023	1.016	Anpassung für Abnahme	Jürgen Lüders
28.11.2023	1.017	Korrekturen und Anpassungen	Andre Katzenberger / Holger Topp
29.11.2023	1.1	Finalisierung	Jürgen Lüders
15.02.2024	1.1.1	Anpassung der Fehlertypen	Moritz Kölblinger
03.04.2024	1.1.2	Anpassung der Fehlertypen	Moritz Kölblinger
10.06.2024	1.1.3	Korrektur der Beschreibung von CmdValidateApp	Moritz Köblinger

Die Viaboxx GmbH hat der go.Rheinland GmbH und dem Zweckverband Nahverkehr Westfalen-Lippe (NWL) dieses Dokument zur Weiterleitung an die in den Verbandsgebieten tätigen Betreiber sowie an deren Lieferanten von Komponenten für Radabstellanlagen (darin eingeschlossen sind Beteiligte, die derartige Anlagen planen bzw. anbieten) für Zwecke des Anschlusses von Fahrradabstellanlagen und das B+R-Buchungs- und Zugangssystems von go.Rheinland/NWL zur Verfügung gestellt.

Inhaltsverzeichnis

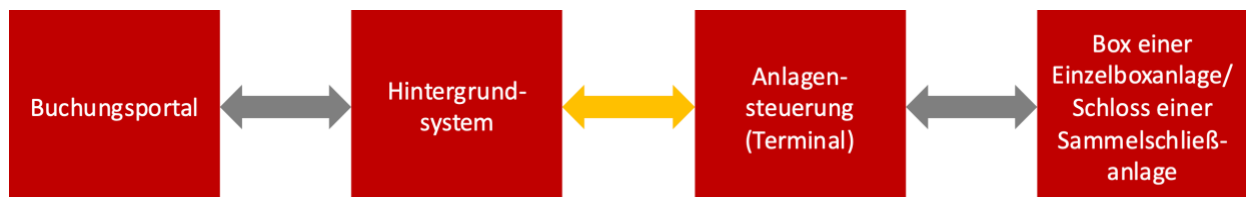
1	<u>EINLEITUNG</u>	<u>4</u>
2	<u>GRUNDLAGEN.....</u>	<u>6</u>
2.1	BEGRIFFE.....	6
2.2	TRANSPORT	7
2.3	INBETRIEBNAHME	8
2.4	BASISDATEN.....	9
2.5	ANTWORTEN.....	9
2.6	SPEICHERUNG.....	11
3	<u>KOMMANDOS FÜR DEN REGELBETRIEB</u>	<u>12</u>
3.1	ANKÜNDIGUNG.....	12
3.2	AUTHENTIFIZIERUNG MIT TAN	15
3.3	AUTHENTIFIZIERUNG MIT RFID-KARTE	16
3.4	AUTHENTIFIZIERUNG UND ÖFFNUNG REMOTE	17
3.5	NEUE AD-HOC-BUCHUNG STARTEN	18
3.6	SPERRUNG / LÖSCHEN EINER RFID-KARTE.....	19
3.7	ÄNDERUNG EINER ZUGANGS-TAN.....	21
3.8	REGISTRIERUNG EINER NEUEN RFID-KARTE	21
4	<u>MONITORING-FUNKTIONEN</u>	<u>24</u>
4.1	TÜRÖFFNUNG.....	24
4.2	STATUSWECHSEL EINER TÜR.....	25
4.3	DATENWIEDERHERSTELLUNG IM TERMINAL.....	26
5	<u>PROZESSBESCHREIBUNG</u>	<u>28</u>
5.1	NUTZER REGISTRIERT SICH AUF RADBOX.NRW	28
5.2	NUTZER BUCHT UND BEGINNT DEN PARKVORGANG.....	28
5.3	EINE BUCHUNG VERLÄNGERN	31
5.4	NUTZER ENTNIMMT SEIN FAHRRAD.....	31
5.5	STORNO EINER BUCHUNG.....	32
5.6	SERVICE-PERSONAL.....	33
5.7	SERVICEPIN VERARBEITEN	33
5.8	EINE NEUE TAN WIRD VERGEBEN	34
5.9	BUCHUNGSDATEN AUF DEM TERMINAL WIEDERHERSTELLEN.....	34
6	<u>STANDARDS FÜR DAS ETICKET</u>	<u>35</u>

1 Einleitung

Die Viabooxx GmbH hat im Auftrag der go.Rheinland GmbH (vormals Nahverkehr Rheinland GmbH (NVR)) und des Zweckverband Nahverkehr Westfalen-Lippe (NWL) das Buchungs- und Zugangssystem (im Folgenden Hintergrundsystem genannt) für gesicherte Fahrradabstellanlagen (nachfolgend Anlagen genannt) in den beiden Verbandsgebieten entwickelt und betreibt dieses System. Wesentliche Bestandteile des Hintergrundsystems sind ein operatives Buchungsportal zur Bedienung aller Kundeninteraktionen (u. a. Informieren über und Suche nach Anlagen/Stellplätzen, Nutzerregistrierung und -verwaltung, Buchen von Stellplätzen zur Nutzung der Anlagen, Zahlungsabwicklung), die Verwaltung von Kundendaten. Systemkonfigurationen von Betreiber im Rahmen vorgegebener Möglichkeiten und weitere Funktionen sowie die Bereitstellung von Schnittstellen zu Drittsystemen (Anlagenhersteller, Buchungsportale) und Statistiken.

Kommunen, Verkehrsbetriebe bzw. Betreiber von Anlagen können neue und bestehende Anlagen an das Hintergrundsystem ankoppeln. Dazu müssen die Anlagen über eine elektronische Schließeinheit je Tor, ein Terminal mit Anzeige, einer Tastatur für die Eingabe der TAN, einem QR-Code-Scanner und RFID/NFC-Leseinheit für Benutzer und einen Internetzugang verfügen, um mit dem Hintergrundsystem von Viabooxx zu kommunizieren.

In diesem Dokument wird die Schnittstelle beschrieben, über die eine Anlage mit dem Hintergrundsystem kommunizieren kann. Im folgenden Bild ist die Lokation der Schnittstelle in Gelb dargestellt – zwischen dem Hintergrundsystem und dem Terminal an der Anlage:



Die interne Steuerung der Anlage ist Aufgabe des Anlagenherstellers.

Die hier beschriebene Schnittstelle dient dazu, einer Anlage mitzuteilen, dass Buchungen vorliegen sowie die Authentifizierung und Autorisierung von Nutzern zu gewährleisten. Die Methoden zum Austausch der Buchungsdaten und zur Steuerung der Anlage sind zwingend erforderlich für die durchgehende Funktionalität des Gesamtsystems. Die konkrete lokale Umsetzung auf dem Terminal entscheidet der Anlagenhersteller, hier wird nur das erwartete Ergebnis beschrieben (z. B. eine bestimmte Tür oder eine Box zu öffnen)

Diese Unterlage stellt die wesentlichen Aspekte der Schnittstelle zusammen und dient Anbietern von Fahrradabstellanlagen als Grundlage einer Aufwandsschätzung und der Umsetzung einer terminalseitigen Kommunikationsschnittstelle.

Die Beschreibung gliedert sich dabei in die folgenden Bereiche:

- **Grundlagen**
Beschreibung der Schnittstellenarchitektur
- **Funktionen im Regelbetrieb**
Beschreibung der Aufrufe und Antworten für alle Funktionen, die die Kundeninteraktion unterstützen
- **Monitoring-Funktionen**
Beschreibung der Funktionen, die für die Überwachung der Anlagen notwendig sind
- **Prozessbeschreibung**
Beschreibung der wichtigsten Prozesse, die in der Software des Terminals implementiert sein sollten

2 Grundlagen

Im nachfolgenden Abschnitt sind Begriffsbestimmungen, sowie Informationen zur Kommunikations-Architektur, zu Basisdaten und zur Messaging-Infrastruktur zu finden.

2.1 Begriffe

Im Folgenden wird der Begriff „Box“ für eine abschließbare Fahrradabstellanlage verwendet. In einer solchen Box können ein oder mehrere „Slots“ (=Stellplätze) für Fahrräder sein. Das bedeutet, dass eine Einzelboxanlage aus einer Gruppe von Boxen mit je einem einzelnen Slot besteht, wohingegen eine Sammelschließanlage aus einer (oder mehreren) Boxen mit jeweils einer Gruppe von Slots für die abzustellenden Räder besteht. Schließanlagen, die ebenfalls an dieses Hintergrundsystem angeschlossen werden können, werden wie Einzelboxen behandelt und stellen eine eigene Anlage dar, auch wenn sich diese Schließfächer innerhalb einer Sammelanlage befinden.

Die im weiteren genannten RFID-Karten fallen in die beiden Kategorien:

- eTicket Karte eines Verkehrsverbundes/einer Tarifgemeinschaft
Bei diesem Kartentyp wird als ID die OrgNummer und ApplInstanznummer übermittelt (Siehe Dokumentation der VDV-Kernapplikation). Das Format wird wie folgt erwartet: OrgID - ApplID (Beispiel: 1234-10987654). Der Bindestrich zwischen OrgID und ApplID muss ggf. zusätzlich von der Terminal-Software gesetzt werden.
- System RFID-Karte
Bei diesem Kartentyp wird die UID der Karte selbst übermittelt. Format in Hexadezimalen Zahlenformat getrennt durch „:“ (Beispiel AF:0E:10:FF). Die Länge (Anzahl Bytes) kann variable sein.
- Grundsätzlich werden die einzelnen Bytes durch: (Doppelpunkt) getrennt, analog zu einer MAC-Adresse.

2.2 Transport

Um Anlagen und das Hintergrundsystem weitgehend getrennt handhaben zu können, setzt die Schnittstelle auf eine asynchrone Kommunikation über Message Queues: Jede Anlage bekommt eindeutig zugeordnete Message Queues zur Kommunikation vom Hintergrundsystem zur Anlage und von der Anlage zum Hintergrundsystem. Dabei wird es in jede Richtung jeweils eine Queue geben:

Serverseitig wird RabbitMQ eingesetzt und als Protokoll AMQP.

Auch wenn die Kommunikation regulär eine Reihenfolge einhält, so muss das Terminal im Stande sein, auch mit Anfragen umzugehen, die sich daran nicht halten – so kann es sein, dass eine Anforderung vom Hintergrundsystem das Terminal erreicht, während das Terminal noch auf die Antwort auf eine eigene Anfrage wartet.

Der Message Queue Server ist Teil des Hintergrundsystems und ist über das Internet erreichbar. So können alle Anlagen mit dem Message Queue Server kommunizieren, ohne dass dem Hintergrundsystem die IP-Adressen einer Anlage bzw. eines Terminals bekannt sein muss. Die Kommunikation ist über TLS gesichert.

Auch wenn eine Kommunikation über Message Queues grundsätzlich asynchron ist, setzt die im Folgenden beschriebene Schnittstelle auf eine direkte Beantwortung von eingehenden Nachrichten. Dieses erlaubt es beiden Kommunikationspartnern die Verfügbarkeit der Gegenseite schnell zu erkennen: Wenn eine Nachricht nach einem kurzen Timeout von maximal 30 Sekunden nicht beantwortet wird, kann die sendende Seite davon ausgehen, dass die Verbindung unterbrochen ist:

- Die Anlage kann versuchen, mit den vorliegenden Daten offline zu arbeiten

- Parallel dazu versucht das Terminal, die Verbindung zum Message Queue Server wieder herzustellen.

Für den Fall des kompletten Datenverlustes auf dem Terminal, zum Beispiel wegen des Austausches von Hardware des Terminals und der Notwendigkeit, die Buchungsdaten neu zu befüllen, stellt die Schnittstelle eine dedizierte Funktion bereit („CmdRestoreBackup“).

Achtung: Dieser Aufruf ist aufgrund möglicher Seiteneffekte und „RaceConditions“ nicht zur Synchronisation im Regelbetrieb geeignet und aus diesem Grund auch unbedingt zu vermeiden.

2.3 Inbetriebnahme

Für den initialen Aufbau der Kommunikation des Terminals mit dem Hintergrundsystem sind für eine erste Inbetriebnahme die Kommunikationsparameter wie IP-Adressen des Message Queue Servers, Portfreigaben etc. auf dem Terminal zu speichern. Die erforderlichen Queues werden auf Anforderung des Terminal-Herstellers von Viabooxx angelegt. Diese Informationen sind aus Sicherheitsgründen nicht Teil dieses Dokumentes und müssen bei Viabooxx angefragt werden.

Bevor eine Anlage live geschaltet werden kann, müssen außerdem vom Hersteller des Terminals Abnahmetests gegen ein Testsystem der Viabooxx durchgeführt werden, um damit die vollständige und korrekte Implementierung der Funktionsfähigkeit zu dokumentieren. Am Ende der Abnahmetests wird ein Protokoll erstellt, das es auch den beauftragenden Anlagenbetreibern (Kommunen, Verkehrsunternehmen) ermöglicht, die (Nicht-) Erfüllung der einzelnen Funktionen nachzuvollziehen.

2.4 Basisdaten

Über die Message Queues werden Nachrichten im JSON Format ausgetauscht. Diese Nachrichten enthalten Standardfelder, die in den meisten Nachricht erscheinen:

- **commandID**
Eindeutige ID des angeforderten Kommandos seitens des absendenden Partners. Diese IDs sind bei den einzelnen Befehlen im folgenden Kapitel aufgeführt.
- **terminalID**
Die terminalID ist die eindeutige Nummer der Anlage im Hintergrundsystem. Sie wird dort bei der Inbetriebnahme der Anlage vergeben. Sie dient beiden Kommunikationspartnern zur Verifikation, dass sie auch wirklich gemeint sind. Sie ist gleich der client-ID des Message Brokers.

In der Beschreibung der Aufrufparameter sind diese beiden Parameter stets vorzusehen.

2.5 Antworten

In der Kommunikation kommen zwei grundsätzliche Antworten zum Tragen:

SimpleResponse

Der überwiegende Teil der Kommunikation erwartet keine komplexen Antworten, sondern lediglich eine einfache Bestätigung und/oder einen Statuscode

Beispiel:

```
{  
  "commandID": <Wie bei Aufruf>,  
  "terminalID": 1234,  
  "statusCode": "OK"  
}
```

ValidationResponse

Im Falle einer Anmeldung durch einen Benutzer werden über diesen Antworttypen alle relevanten Informationen zu dem Benutzer und seiner Buchung bzw. Berechtigung (bei Technikern und Reinigungskräfte) übermittelt.

Beispiel:

```
{
  "commandID": <Wie bei Aufruf>,
  "terminalID": 1234,
  "userTAN": "123456",
  "slots": [3],
  "statusCode": "OK"
}
```

Die beiden Antworttypen enthalten den Parameter „statusCode“. Dieser kann die folgenden Werte annehmen:

Code	Bedeutung
OK	Alles OK, keine Fehler
ErrorTerminalWrong	Die übermittelte terminalID stimmt nicht überein. Konfigurationsfehler.
ErrorTANUnknown	Bei der Authentifizierung über eine TAN Eingabe: TAN ist falsch oder unbekannt.
ErrorCardWrong	Bei der Authentifizierung über eine RFID-Karte: Diese Karte ist falsch oder unbekannt.
ErrorBoxUnknown	Die in der Anfrage enthaltene Box-Nummer ist der Anlage unbekannt.
ErrorSlotsUnknown	Die in der Anfrage enthaltenen Slot-Nummern sind der Anlage unbekannt.
ErrorNoReservation	Die Authentifizierung des Nutzers hat funktioniert – aber an dieser Anlage liegt keine Buchung für den Benutzer vor.
ErrorCardUnknown	Bei der Sperrung einer RFID-Karte meldet das Terminal, dass es diese Karte nicht kennt.

ErrorTANWrong	Bei der Änderung einer Benutzer TAN durch das System, meldet das Terminal, dass es diese TAN nicht kennt.
ErrorCardIdNotValid	Bei dem Versuch, eine neue RFID-Karte zu registrieren, meldet das Terminal, dass es diesen Kartentyp nicht unterstützt.
ErrorUnsupported	Funktion wird nicht unterstützt: Wenn das Terminal bestimmte Funktionen hardwaremäßig nicht unterstützt.
ErrorNoSpace	Eine ad Hoc Buchung wurde initiiert, es gibt aber keine freien Plätze mehr in der Anlage.
ErrorUnknown	Ein nicht weiter oben spezifizierter Fehlercode als Fallback.

Die Übertragung des Status erfolgt gemäß der aufgeführten Tabelle. Um eine einheitliche Information des Nutzers über detektierte Fehlerzustände in den verschiedenen Komponenten des gesamten Systems zu gewährleisten, werden die dem Nutzer bereitgestellten Antwort-Texte im Terminal in einer Tabelle mindestens in deutscher Sprache vorgehalten. Die Tabelle wird von Viaboxx als Messagebundle mit aktuellem Inhalt nach Rückfrage bereitgestellt.

2.6 Speicherung

Viele Informationen müssen auf dem Terminal lokal gespeichert werden. Damit erlaubt das Terminal auch eine Benutzerinteraktion, wenn die Internetverbindung unterbrochen ist. In der Beschreibung der Kommandos sind diese Daten explizit erwähnt.

Da in der Schnittstelle Daten für das Terminal lesbar übertragen werden, sollte das Terminal geeignete Maßnahmen ergreifen, die Daten vor einer lokalen Speicherung entsprechend zu verschlüsseln. Die Nutzung von Hashing-Verfahren ist dabei nur dann ausreichend, wenn der genutzte Algorithmus nicht anfällig für Brute-Force Attacken ist.

3 Kommandos für den Regelbetrieb

Im Folgenden sind alle Kommandos aufgeführt, die für die reguläre Benutzerinteraktion notwendig sind. Die Kommandos und Events des Terminals führen im Hintergrundsystem zu einer konsistenten Datenhaltung, sodass diese erforderlich für die Funktionsweise des Gesamtsystems sind.

Wenn das Terminal und das Hintergrundsystem durch einen Ausfall der Netzwerkverbindung nicht miteinander kommunizieren können, so müssen die aufgeführten Nachrichten am Terminal so lange lokal gespeichert werden, bis die Verbindung zum Messageserver wieder etabliert ist, um dann alle ausstehenden Messages zu versenden und solche vom Hintergrundsystem zu empfangen. Die Nachrichten werden nach der Wiederherstellung der Verbindung dann vom jeweils Empfangenden verarbeitet, sodass Veränderungen am Terminal oder neue Buchungen auf dem Hintergrundsystem nachträglich an den jeweils anderen Kommunikationspartner übertragen werden. Das Terminal ist in der Kommunikation seines Status führend, das Hintergrundsystem übernimmt diesen Status zur Anzeige der Verfügbarkeiten von Stellplätzen.

3.1 Ankündigung

Wenn ein Nutzer über das Buchungsportal eine Anlage ausgesucht und einen Platz gebucht hat, sendet das Hintergrundsystem einen Buchungsdatensatz mit allen dazugehörenden Daten an das betroffene Terminal. Somit liegen alle Daten bereits auf dem Terminal vor, um bei einem temporären Offline-Fall, den Benutzer autark bedienen zu können und ihm eine Öffnung der Anlage zu erlauben. Dies setzt entsprechende Speicherkapazitäten und Prozessimplementierungen im Terminal voraus.

Nach dem Ende der Buchung (also sobald das Ende der Buchungszeit abgelaufen ist) können die Zugangsdaten vom Terminal gelöscht werden.

Ausgehend:

Eine Ankündigung wird vom Hintergrundsystem an das Terminal gesendet.

Beschreibung der Parameter:

Name	Wert/Bedeutung	Speichern
commandID	„CmdNotification“ – Buchungsinformation	
terminalID	ID des Terminals	
bookingID	ID der zugehörigen Buchung Es gilt immer der zuletzt gesendete Status, sodass mit dieser Nachricht sowohl neue Buchungen als auch Änderungen auf bestehende Buchungen übertragen werden.	Ja
version	Aktuelle Version der Buchung, der Wert wird serverseitig inkrementiert. Das Terminal kann damit erkennen, dass der gesendete Datensatz veraltet oder aktuell ist.	Ja
start	Timestamp des Buchungsstarts mit Datum und Uhrzeit. Vor diesem Timestamp ist die Nutzung der Anlage durch den Benutzer nicht möglich. Der Timestamp entspricht im Format dem ISO-Standard (ISO 8601) mit Uhrzeit in UTC. Beispiel: „2021-06-24T18:21:00,000Z“ Z steht hier für die Zeitzone (Z=UTC). Weitere Details hierzu sind unter https://de.wikipedia.org/wiki/ISO_8601 zu finden.	Ja
end	Timestamp des Nutzungsendes mit Datum und Uhrzeit. Nach diesem Timestamp ist eine Nutzung der Anlage durch den Benutzer nicht möglich. Das Format entspricht dem wie bei „start“	Ja
slots	Liste von Stellplätzen, die zu dieser Buchung gehören. Mehrere Stellplätze werden innerhalb der eckigen Klammern durch Komma getrennt angegeben (siehe JSON-Spezifikation)	Ja

Name	Wert/Bedeutung	Speichern
userTAN	TAN für die Buchung des Nutzers	Ja
cards	Liste von Karten, um die registrierten Karten des Benutzers zu identifizieren. Eine Karte besteht aus Karten-ID und Kartentyp cardID: Bei einer eTicket Karte die Kombination aus OrgNummer und AppInstanznummer Bei einer Systemkarte die UID der Karte cardType: 1: eTicket 2: System RFID-Karte card : { cardID : <value>, cardType : <value> }	Ja

Antworttyp: SimpleResponse

Mögliche Fehler:

- ErrorTerminalWrong
- ErrorSlotsUnknown

Beispiel:

Ein Stellplatz (z. B. Nummer 12) vom 03.09.2021 bis zum 10.09.21 mit einer TAN und einer eTicket Karte

```
{
  "commandID": "CmdNotification",
  "terminalID": 1234,
  "bookingID": 123,
  "version": 1,
  "start": "2021-09-03T-09:00:00.000Z",
```

```

"end": "2021-09-10T-08:59:59.999Z",
"slots": [12],
"userTAN": "123456",
"cards": [
  { "cardID": "3886-00250001", cardType: 1},
  { "cardID": "1234-99220002", cardType: 1}
]
}

```

3.2 Authentifizierung mit TAN

Ein Nutzer kann sich mit der TAN authentifizieren, die er bei der Buchung erhalten hat. Dabei ist es nicht relevant, ob die TAN über ein Tastenfeld eingegeben wurde, oder als QR-Code eingelesen wurde. Diese Funktion kann genutzt werden, um die TAN eines Nutzers zu validieren und muss für die Validierung von Service-PINS, die an Reinigungskräfte, Techniker und Administratoren vergeben wurden, genutzt werden, da für diese Gruppe von Nutzern kein *CmdNotification* gesendet wird.

Ausgehend:

Das Terminal sendet die Authentifizierungsanfrage an das Hintergrundsystem.

Beschreibung der Parameter:

Name	Wert/Bedeutung
commandID	„CmdValidateTAN“ – Authentifizierung mit TAN
terminalID	ID des Terminals
userTAN	TAN des Benutzers

Antworttyp: ValidationResponse

Mögliche Fehler:

- ErrorTANUnknown
- ErrorTerminalWrong

Beispiel:

Der Kunde hat als TAN die 67342 eingegeben:

```
{
  "commandID": "CmdValidateTAN",
  "terminalID": 1234,
  "userTAN": "67342"
}
```

3.3 Authentifizierung mit RFID-Karte

Ein Nutzer kann sich mit der RFID-Karte authentifizieren, die er in seinem Benutzerprofil hinterlegt hat, bzw. die bei einer früheren Buchung über das Terminal in das Benutzerprofil übertragen wurde (s. Kap. 3.8).

Ausgehend:

Das Terminal sendet die Authentifizierungsanfrage an das Hintergrundsystem.

Beschreibung der Parameter:

Name	Wert/Bedeutung
commandID	„CmdValidateRFID“ – Authentifizierung mit RFID-Karte
terminalID	ID des Terminals
cardID	Bei einer eTicket Karte die Kombination aus OrgNummer und ApplInstanznummer Bei einer Systemkarte die UID der Karte
cardType	1: eTicket 2: System RFID-Karte

Antworttyp: ValidationResponse

Mögliche Fehler:

- ErrorCardUnknown
- ErrorTerminalWrong

Beispiel:

Der Kunde hat eine eTicket Karte genutzt:

```
{
```



```

"commandID": "CmdValidateRFID",
"terminalID": 1234,
"cardID": "3886-00250001",
"cardType": 1
}

```

3.4 Authentifizierung und Öffnung Remote

Ein Nutzer kann, z. B. über eine App, die Öffnung seiner Box initiieren. Bei dieser Öffnungsmöglichkeit ist die Anzahl der maximal zu nutzenden Stellplätze fachlich definiert auf Eins (1) beschränkt, da via App nur genau eine Box pro Aufruf geöffnet werden darf, d.h. bei Sammelanlagen mit einem Zugang für mehrere Slots ist diese Zugangsoption im Hintergrundsystem deaktiviert.

Ausgehend:

Das Hintergrundsystem sendet diese Anfrage an das Terminal.

Beschreibung der Parameter:

Name	Wert/Bedeutung	Speichern
commandID	„CmdValidateApp“ – Authentifizierung und Öffnung Remote	
terminalID	ID des Terminals	
userId	Id des Users	
bookingId	Id der Buchung	
bookingTan	TAN des Benutzers als Backup für den Zugang	Ja
slots	Liste von Stellplätzen, die zu dieser Buchung gehören. In diesem Fall ist die Länge der Liste auf eins beschränkt.	Ja
status	„OK“ oder „ErrorUnknown“ (historisches Feld – kann ignoriert werden)	
boxId	Id der Box	

Antworttyp: SimpleResponse

Mögliche Fehler:

- ErrorTerminalWrong
- ErrorSlotsUnknown

Beispiel:

Der Kunde hat als TAN die 123456:

```
{
  "commandID": "CmdValidateApp",
  "terminalID": 1234,
  "userId": "f88042a9-8abc-4fff-9c60-e00d3849910f",
  "bookingId": 123,
  "bookingTan": "123456",
  "slots": [3],
  "status": "OK",
  "boxId": 123
}
```

3.5 Neue Ad-hoc-Buchung starten



Die Funktion Ad-hoc-Buchung wird vom Hintergrundsystem ab Version 1.1 nicht mehr unterstützt. Die Funktion lässt sich weiterhin aufrufen führt aber zu keinem der hier beschriebenen Prozessen / Vorgängen.

Ein Kunde kann, wenn er eine RFID-Karte in seinem Profil gespeichert hat, auch ohne vorhergehende Buchung einen Stellplatz bekommen. Dazu identifiziert er sich mit der RFID-Karte. Wenn in der Anlage noch Platz für ein Rad an dem Tag frei ist, bekommt er vom Hintergrundsystem die Zugangsdaten und das Terminal kann ihm diese anzeigen.

Achtung: Diese Funktion setzt den Online-Status der Anlage voraus und darf nur dann aufgerufen werden, sodass die Buchung sicher im Hintergrundsystem eingetragen wird.

Ausgehend:

Das Terminal sendet diese Anfrage an das Hintergrundsystem.

Beschreibung der Parameter:

Name	Wert/Bedeutung
commandID	„CmdBooking“ – Ad Hoc Buchung
terminalID	ID des Terminals
cardID	ID der Karte: Bei einer eTicket Karte die Org-Nummer und ApplikationsInstanz-Nummer Bei einer Systemkarte die UID der Karte
cardType	1: eTicket 2: System RFID-Karte

Antworttyp: ValidationResponse

Mögliche Fehler:

- ErrorCardUnknown
- ErrorNoSpace

Beispiel:

Der Kunde hat eine RFID-Karte:

```
{
  "commandID": "CmdBooking",
  "terminalID": 1234,
  "cardID": "3386-00250001",
  "cardType": 1
}
```

3.6 Sperrung / Löschen einer RFID-Karte

Wenn ein Nutzer seine RFID-Karte verliert, kann er dieser Karte in seinem Kundenprofil im Hintergrundsystem löschen. Anschließend sendet das Hintergrundsystem eine CmdBlockRFID an alle Terminals. Jedes empfangende Terminal sollte diese Karte dann in den aktuellen Belegungen suchen und ggf. dort ebenfalls löschen. Die Buchung muss jedoch bestehen bleiben, nur der Zugang mit der RFID-Karte ist nicht mehr zu gewähren. Gleichwohl kann die

cardID zu einem späteren Zeitpunkt wieder im Profil eines Nutzers hinterlegt werden, die cardID sollte also nicht in einer internen Black-List geführt werden und somit auf Seiten der Terminal-Software generell für die Zukunft gesperrt werden.

Ausgehend:

Das Hintergrundsystem sendet diese Anfrage an das Terminal.

Beschreibung der Parameter:

Name	Wert/Bedeutung	Speichern
commandID	„CmdBlockRFID“ – Sperrung einer RFID-Karte	
terminalID	ID des Terminals	
cardID	ID der zu löschenden Karte: Bei einer eTicket Karte die Kombination aus OrgNummer und ApplInstanznummer Bei einer Systemkarte die UID der Karte	Ja – durch Löschen des alten Eintrages
cardType	1: eTicket 2: System RFID-Karte	

Antworttyp: SimpleResponse

Mögliche Fehler:

- ErrorTerminalWrong
- ErrorCardUnknown

Beispiel:

```
{
  "commandID": "CmdBooking",
  "terminalID": 1234,
  "cardID": "3386-00250001",
  "cardType": 1
}
```

3.7 Änderung einer Zugangs-TAN

Ein Kunde kann die TAN, die zu seiner Buchung gehört, über das Hintergrundsystem ändern lassen. Hierzu sendet das Hintergrundsystem die alte und die neue TAN an das Terminal, an dem die TAN genutzt werden kann. Das empfangende Terminal sollte diese TAN dann in den aktuellen Belegungen suchen und ggf. dort ändern.

Ausgehend:

Das Hintergrundsystem sendet diese Anfrage an das Terminal.

Beschreibung der Parameter:

Name	Wert/Bedeutung	Speichern
commandID	„CmdNewTAN“ – Änderung einer TAN	
oldUserTAN	Die alte TAN	
newUserTAN	Die neue TAN	Ja

Antworttyp: SimpleResponse

Mögliche Fehler:

- ErrorTerminalWrong
- ErrorTANUnknown

Beispiel:

Der Kunde hat als alte TAN die 123456 und als neue TAN die 654321:

```
{
  "commandID": "CmdNewTAN",
  "oldUserTAN": "123456",
  "newUserTAN": "654321"
}
```

3.8 Registrierung einer neuen RFID-Karte

Ein Kunde kann am Terminal seine RFID-Karte (nur eTicket oder System RFID-Karte) registrieren und in seinem Profil speichern. Dazu ist es notwendig, an der

Anlage eine laufende Buchung zu haben, um sich damit zu authentifizieren. Diese Karte kann nach der Registrierung für den Zugang an den Anlagen für zukünftige Buchungen verwendet werden (siehe CmdNotification), für bestehende Buchungen kann die Karte nicht verwendet werden.

Das Terminal kann sowohl eTickets als auch System RFID-Karten differenziert "erkennen" und sendet diese Anfrage an das Hintergrundsystem.

Ausgehend:

Das Terminal sendet diese Anfrage an das Hintergrundsystem.

Beschreibung der Parameter:

Name	Wert/Bedeutung
commandID	„CmdNewCard“ – Registrierung einer neuen Karte
terminalID	ID des Terminals
userTAN	Valide TAN einer laufenden Buchung an dieser Anlage
cardID	ID der zu registrierenden Karte: Bei einer eTicket Karte die Kombination aus OrgNummer und ApplInstanznummer Bei einer Systemkarte die UID der Karte
cardType	1: eTicket 2: System RFID-Karte

Antworttyp: SimpleResponse

Mögliche Fehler:

- ErrorTerminalWrong
- ErrorTANUnknown
- ErrorCardIdNotValid
- ErrorUnknown

Beispiel:

Der Kunde hat als TAN die 123456:

```
{  
  "commandID": "CmdNewCard",  
  "terminalID": 1234,  
  "userTAN": "123456",  
  "cardID": "3386-00250001",  
  "cardType": 1  
}
```

4 Monitoring-Funktionen

4.1 Türöffnung

Wenn sich die Tür einer Box öffnet, wird dies vom Terminal an das Hintergrundsystem gemeldet.

Ausgehend:

Das Terminal sendet an das Hintergrundsystem.

Beschreibung der Parameter:

Name	Wert/Bedeutung
commandID	EventDoorOpened – Türe wurde geöffnet
terminalID	ID des Terminals
boxID	ID der geöffneten Tür
slots	Liste der nun erreichbaren Stellplätze
userTAN	Nur in der Nachricht, wenn die Öffnung durch die Eingabe einer TAN erfolgt ist
cardID	Nur in der Nachricht, wenn die Öffnung durch eine RFID-Karte erfolgt ist
cardType	1: eTicket 2: System RFID-Karte
reasonType	1: durch einen Nutzer 2: durch Wartung/Techniker/Betreiber 3: Notöffnung 4: sonstiges

Antworttyp: SimpleResponse

Mögliche Fehler:

- ErrorBoxUnknown

- ErrorTerminalWrong
- ErrorUnknown

Beispiel:

Die Tür der Box 3 wurde durch einen Nutzer mit einer gültigen Buchung geöffnet, sie gehört zu einer Sammelschließanlage mit 6 Plätzen hinter dieser Türe:

```
{
  "commandID": "EventDoorOpened",
  "terminalID": 1234,
  "boxID": 3,
  "slots": [7, 8, 9, 10, 11, 12],
  "userTAN": "123456",
  "reasonType": 1
}
```

4.2 Statuswechsel einer Tür

Wenn das Terminal versucht hat, eine Box zu öffnen, diese aber nicht geöffnet werden konnte oder die Tür noch aufsteht, wird dies mit einem Fehlerstatus an das Hintergrundsystem zurückgemeldet. Sobald die Tür wieder verwendbar ist, sendet das Terminal eine Nachricht mit „OK“, sodass das Hintergrundsystem diese Box wieder für weitere Buchungsvorgänge anbieten kann.

Jede Information, die das Terminal sendet, ist hierbei führend. Auch regelmäßige Statusmeldungen der Terminals sind möglich, das Hintergrundsystem aktualisiert sich dann jeweils mit den neuesten Informationen.

Ausgehend:

Das Terminal sendet an das Hintergrundsystem.

Beschreibung der Parameter:

Name	Wert/Bedeutung
commandID	„EventDoorStatus“ – Nutzungsstatus einer Tür
terminalID	ID des Terminals
boxID	ID der gemeldeten Tür

status	Ein numerischer Code, der den Status der Tür beschreibt. Alle Codes > 0 sind dabei Fehler und werden durch das Hintergrundsystem gleich behandelt. 0: OK 1: Tür defekt 2: Tür offen 3: Sonstiger Fehler
--------	--

Antworttyp: SimpleResponse

Mögliche Fehler:

- ErrorBoxUnknown
- ErrorTerminalWrong
- ErrorUnknown

Beispiel:

Die Tür der Box 3 ließ sich nicht mehr öffnen:

```
{
  "commandID": "EventDoorStatus",
  "terminalID": 1234,
  "boxID": 3,
  "Status": 1
}
```

4.3 Datenwiederherstellung im Terminal

Wenn das Terminal durch Neuaufsetzen oder sonstigen Datenverlust keine Buchungsdaten mehr gespeichert hat, so kann es beim Hintergrundsystem eine Anfrage stellen, um alle laufenden Buchungen zur Speicherung neu zu erhalten. Übertragen werden diese Einzelbuchungen als Ankündigungen („*CmdNotification*“) mit den aktuell gespeicherten Informationen im Hintergrundsystem.

Ausgehend:

Das Terminal sendet an das Hintergrundsystem.

Beschreibung der Parameter:

Name	Wert/Bedeutung
commandID	„CmdRestoreBackup“ – Anfrage zur Datenwiederherstellung
terminalID	ID des Terminals

Antworttyp: SimpleResponse

Außerdem *n* Nachrichten „CmdNotification“

Mögliche Fehler:

- ErrorTerminalWrong
- ErrorUnknown

Beispiel:

```
{
  "commandID": "CmdRestoreBackup",
  "terminalID": 1234
}
```

5 Prozessbeschreibung

Nachfolgend werden die wichtigsten Prozesse bei der Nutzung einer Abstellanlage beschrieben, um ein Verständnis zu entwickeln, welche Prozesse auf dem Terminal eine Anlage erforderlich sind und wie diese ggf. gestaltet werden müssen.

5.1 Nutzer registriert sich auf radbox.nrw

Für ein besseres Verständnis, wie die Terminal-Schnittstelle zu radbox.nrw zu nutzen ist, ist nachfolgend beschrieben, wie ein Kunde und damit Nutzer sich auf radbox.nrw registriert und welche Daten im Rahmen der Registrierung erhoben und auf dem radbox.nrw-Server gespeichert werden. radbox.nrw wird mittels Browser aufgerufen und kann auf einem Rechner, Tablet-Computer oder Smartphone aufgerufen werden.

1. Nutzer legt neuen Account in radbox.nrw an.
2. Nutzer hinterlegt dazu E-Mail-Adresse und vergibt ein Passwort für den Zugriff auf sein Konto
3. Nutzer hinterlegt auch eine Antwort auf eine Sicherheitsfrage.
4. Nutzer hinterlegt ein Zahlungsmittel, von dem künftig das Entgelt für die Parkvorgänge abgebucht wird
5. Nutzer hinterlegt optional ein RFID-Zugangsmedium (eTicket oder System RFID-Karte), mit dem sich der Nutzer an Terminal/Anlagen anmeldet, um zum Beispiel einen Parkvorgang zu beginnen. Dies trifft nur für solche Terminals zu, die mit einer Leseeinrichtung für RFID-Karten ausgestattet sind.

5.2 Nutzer bucht und beginnt den Parkvorgang

Nachfolgend ist eine kurze Beschreibung zu finden, wie die Buchung eines Stellplatzes in radbox.nrw abläuft:

1. Nutzer wählt Standort aus der Liste oder über die Landkarte aus.
2. Nutzer wählt Parkdauer aus und den Startzeitpunkt des Parkvorgangs
3. Nutzer wählt einen Stellplatz aus und bestätigt die Buchung
4. Das hinterlegte Zahlungsmittel des Nutzers wird belastet
5. radbox.nrw-Server sendet eine E-Mail mit den Buchungsdaten (ohne TAN) an Nutzer
6. radbox.nrw-Server sendet *CmdNotification* an das entsprechende Terminal
7. Das Terminal bestätigt den Eingang der Buchung mit *SimpleResponse*

Alle für die Buchung und den Zutritt zur Anlage relevanten Daten werden kurz nach einer Buchung auf dem radbox.nrw-Server an das Terminal der Anlage (siehe *CmdNotification*) gesendet. Somit liegen die Daten zur Buchung rechtzeitig an der Anlage vor. Es setzt jedoch voraus, dass die Daten lokal auf dem Terminal gespeichert werden. Aus Sicherheitsgründen ist bei der Implementierung der Terminal-Software darauf zu achten, dass diese Daten verschlüsselt auf dem Terminal gespeichert werden, um eine missbräuchliche Nutzung der Daten zu verhindern/erschweren.

Trifft der Nutzer an der Anlage ein, um den Parkvorgang zu starten, können die Nutzereingaben für eine Buchung gegen die lokal auf dem Terminal gespeicherten Daten, validiert werden. Somit kann auch ein Zutritt in die Anlage gewährt werden, obwohl vorübergehend keine Verbindung zum radbox.nrw-Server besteht (z. B. beim Ausfall der Internet-Verbindung). Gleiches gilt auch, wenn der Nutzer sein geparktes Fahrrad aus der Anlage entnehmen möchte.

Im Hintergrundsystem von radbox.nrw wird dem Wunsch des Anlagenbetreibers entsprechend hinterlegt, wie viele zeitgleiche Buchungen je Nutzer möglich sind.

5.2.1 Medien für den Start des Parkvorgangs

Zum Starten eines Parkvorgangs an der Anlage bestehen grundsätzlich die Möglichkeit nachfolgende vier Medien zu verwenden:

1. TAN-Code
2. QR-Code
3. RFID-Karte
4. Remote-Funktion in der Web-App

Zu den einzelnen Optionen:

1. TAN-Code

Zum Starten des Parkvorgangs gibt der Nutzer am Terminal über eine Tastatur die TAN ein. Die TAN wird dem Nutzer in der Web-App mitgeteilt/angezeigt.

2. QR-Code

Alternativ dazu kann der Nutzer die TAN auch als QR-Code durch das Terminal einlesen lassen. Das setzt voraus, dass das Terminal mit einem entsprechenden Lesegerät ausgestattet ist. Den QR-Code findet der Nutzer in der Web-App bei seinen Buchungsdaten.

3. RFID-Karte

Ein Nutzer kann seinen Parkvorgang am Terminal auch mit einem, in seinem Profil hinterlegten, RFID-Zugangsmedium starten. Dazu hält der Nutzer diese RFID-Karte an das Lesegerät des Terminals, es wird die UID (System RFID-Karte) bzw. OrgID – AppID (bei eTicket) gelesen und danach sucht das Terminal aus den lokal gespeicherten Buchungsdaten die Buchung heraus, bei der mit dem Kommando *CmdNotification* die entsprechende UID bzw. OrgID – AppID mitgesendet wurde.

Falls der Nutzer noch keine RFID-Karte in seinem Profil auf dem Hintergrundsystem hinterlegt hat, kann im Fall einer bestehenden Buchung eine RFID-Karte am Terminal gelesen werden, die UID bzw. eTicket-Nummer wird zusammen mit dem Kartentyp vom Terminal in diesem Fall an das

Hintergrundsystem gesendet, in das Benutzerprofil geschrieben und steht dann für künftige Buchungen zur Öffnung zur Verfügung. (vgl. Kap. 3.8).

4. Remote-Öffnung

Im Falle einer Einzelbox- oder Schließfach-Anlage kann ein Nutzer auch die Öffnung der Tür zu seiner gebuchten Box remote über die WebApp des radbox.nrw-Servers veranlassen. Der Server sendet in diesem Fall ein *CmdValidateApp*-Kommando an das entsprechende Terminal. Das Terminal vermerkt mit dem Empfang dieses Kommandos den Startzeitpunkt des Parkvorgangs (bzw. der Türöffnung) in seinem lokalen Daten und öffnet die entsprechende Tür der Box. Abschließend bestätigt das Terminal diesen Vorgang mit einer *SimpleResponse*-Antwort an den Server. Die Box kann remote auch mehrfach geöffnet werden, z. B. weil ein Nutzer in seiner Box Sachen ablegen möchte. Das Öffnen per Remote-Kommando steht nur für Einzelbox-Anlagen zur Verfügung.

In der Systemverwaltung des Hintergrundsystems wird hinterlegt, welche der o. a. Zugangsoptionen vorhanden ist bzw. genutzt werden kann/soll.

5.3 Eine Buchung verlängern

Ein Nutzer kann über radbox.nrw eine Buchung verlängern. Nachdem der Nutzer die Buchung verlängert hat, sendet das Hintergrundsystem eine *CmdNotification* Kommando an das Terminal. Dabei wird der Parameter „version“ inkrementiert und die Parameter „start“ und „end“ entsprechend aktualisiert.

5.4 Nutzer entnimmt sein Fahrrad

Bei einer Abholung eines Fahrrades bzw. Gepäck in einem Schließfach durch den Nutzer, können die Eingaben des Nutzers am Terminal gegen die lokal auf dem Terminal gespeicherten Daten zur Buchung überprüft werden. So kann

dem Nutzer der Zugriff auf sein Fahrrad gewährt werden, auch wenn vorübergehend keine Internet-Verbindung zum radbox.nrw-Server besteht.

Der Nutzer kann zur Öffnung für die letzte Entnahme seines Fahrrades folgende Medien nutzen (Einschränkungen abhängig vom Anlagentyp):

1. TAN-Code
2. QR-Code
3. RFID-Karte
4. Remote-Funktion in der Web-App

Die Funktionsweise bzw. der Einsatz der verschiedenen Medien erfolgt analog zu „Beginn eines Parkvorgangs“.

Eine aktuelle, gültige und laufende Buchung kann nicht vorzeitig beendet werden. Die Laufzeit des Buchungsvorgangs ist durch die Buchungsdaten, die mittels der Funktion *CmdNotification* festgelegt wurde, fix. Holt der Nutzer sein Fahrrad früher ab, bleibt der Stellplatz bis zum Ablauf der Buchung ungenutzt. Fahrräder, die nicht rechtzeitig abgeholt werden (also über das Buchungsende hinaus), verbleiben in der Fahrradabstellanlage. Der Nutzer hat jedoch die Möglichkeit seine Buchung zu verlängern. Der Nutzer wird rechtzeitig per E-Mail an das Ende der Buchung erinnert, so dass ausreichend Zeit bleibt, um die Buchung zu verlängern oder das Fahrrad rechtzeitig abzuholen.

5.5 Storno einer Buchung

Eine Buchung kann auch durch den Support oder Administratoren des Hintergrundsystems beendet und storniert werden. In diesem Fall sendet das Hintergrundsystem eine *CmdNotification* Message an das Terminal. Dabei wird der Parameter „version“ inkrementiert und die Parameter „start“ und „end“ entsprechend aktualisiert. Da beide Werte in der Vergangenheit liegen, kann die Buchung somit gelöscht werden.

Eine Buchung, deren zeitliche Gültigkeit noch nicht begonnen hat (Zukunftsbuchung oder Verlängerung), kann auch durch den Nutzer storniert werden.

5.6 Service-Personal

Über das Hintergrundsystem radbox.nrw können die Administratoren und Anlagenbetreiber Service-PINs für Service-Personal generieren. Service-Personal (Reinigungskräfte oder Techniker) kann sich durch eine Service-TAN bzw. -PIN am Terminal mit der nachfolgenden Funktion authentifizieren.

- CmdValidateTAN

die vom Terminal an das Hintergrundsystem gesendet werden muss. Das Hintergrundsystem antwortet mit der Message

- ValidationResponse

wenn der Nutzer gültige Zugangsdaten eingegeben hat bzw. mit

- ErrorTANUnknown

wenn die Zugangsdaten nicht gültig sind.

Zurzeit ist eine Prüfung der Service-TAN nur online möglich.

5.7 ServicePIN verarbeiten

ServicePIN werden an Personen vergeben, die die Anlage warten oder reinigen sollen. Diese PIN berechtigen dazu:

- Alle Türen zu öffnen
- Nur Türen zu öffnen die zu leeren Slots / Boxen führen
- Keine Tür öffnen

Die Berechtigung von ServicePIN kann nur online abgefragt werden (siehe *CmdValidateTAN*). Eine ServicePIN kann auch als QR-Code vorliegen und muss auf dem Terminal entsprechend eingelesen werden können.

Am Terminal der Anlage ist ein entsprechender Dialog vorzusehen, über den abgefragt wird, ob mehrere Türen mit Berechtigung oder nur einzelne Türen aus der Gruppe der Berechtigung im Rahmen eines Service-Vorgangs zu öffnen sind. Dieser Dialog sollte mit einem Timeout bzw. Idle-Time versehen sein. Nach Ablauf des Timeout bzw. der Idle-Time sollte der Dialog automatisch geschlossen werden, um zu verhindern das unberechtigte Personen den Dialog nutzen.

Eine Verarbeitung von ServicePIN in Verbindung mit RFID-Karten ist derzeit nicht möglich.

5.8 Eine neue TAN wird vergeben

Kann der Nutzer nicht auf die TAN seiner Buchung zurückgreifen (z. B. Smartphone defekt) kann der Support dem Nutzer einen neue TAN bereitstellen. Nachdem die neue TAN vergeben wurde, sendet das Hintergrundsystem eine *CmdNewTAN*-Kommando an das Terminal. In der Message sind die alte TAN (zum Auffinden der Buchung im lokalen Datenbestand des Terminals) und die neue TAN gespeichert.

5.9 Buchungsdaten auf dem Terminal wiederherstellen

Mit dem Kommando *CmdRestoreBackup* können alle Daten zu den laufenden Buchungen des Terminals vom Server angefordert werden (natürlich nur die Daten, die auch das anfordernde Terminal betreffen).

Das Kommando *CmdRestoreBackup* ist kein Ersatz für das Persistieren der Daten auf dem Terminal, da sich sonst die Menge der versendeten Daten deutlich erhöhen würde und somit die Leistungsfähigkeit des Gesamtsystems deutlich einschränken würde.

Eine durchgängige Offline-Fähigkeit des Terminals wird durch *radbox.nrw* nicht unterstützt.

6 Standards für das eTicket

Die Normen die ein RFID – Kartenleser erfüllen muss, um eTickets lesen zu können:

1. ISO 7816-Konformität
2. Unterstützung sowohl der kontaktbehafteten als auch der kontaktlosen Schnittstelle gemäß ISO 14443 Teil 1-4

Stichwortverzeichnis

Abholung	31
Abnahmetests	8
Ad-hoc-Buchung.....	18
AMQP	7
Änderung einer TAN.....	21
Ankündigung einer Buchung.....	<i>Siehe CmdNotification</i>
Anlagen	4
asynchrone Kommunikation	<i>Siehe Kommunikation</i>
Authentifizierung mit RFID.....	16
Authentifizierung mit TAN	15
Box.....	6
Boxen.....	<i>Siehe Box</i>
Buchung	28
Buchung verlängern	31
Buchungsdaten wiederherstellen	34
Buchungsportal	4
cardID	14
cardType.....	14
CmdBlockRFID	20
CmdNewCard	22
CmdNewTAN	21
CmdNotification	13, 27
CmdRestoreBackup	8, 27
commandID.....	9
Datenverlust	8
Datenwiederherstellung	26
E-Mail-Adresse	28
ErrorCardIdNotValid	11, 22
ErrorCardUnknown	10, 16, 19, 20
ErrorCardWrong	10
ErrorNoReservation.....	10
ErrorNoSpace	11, 19
ErrorSlotsUnknown.....	10, 18
ErrorTANUnknown.....	10, 15, 22
ErrorTANWrong.....	11
ErrorTerminalWrong.....	10, 18, 20, 22
ErrorUnknown	11
ErrorUnsupported	11
eTicket.....	6
EventDoorOpened	24
Fahrradabstellanlagen.....	4
Funktionsfähigkeit	8
Hintergrundsystem	4
Inbetriebnahme	8

IP-Adresse	8
JSON	9
Kommando	12
Kommunikation	7
lokale Daten	11
Message Queue	9
Message Queue Server	7
Message Queue Servers	8
Message Queues	7
Messagebundle	11
Nachrichten	9
Passwort	28
Port	8
Prozessbeschreibung	28
QR-Code	30
Queue	<i>Siehe Message Queue</i>
RabbitMQ	7
<i>RaceConditions</i>	8
Regelbetrieb	12
Registrierung	28
Registrierung RFID-Karte	21
Remote	17
Remote-Öffnung	31
RFID	6, 16, 18, 19, 21
RFID-Karte	30
RFID-Zugangsmedium	28
Service-Personal	33
ServicePIN	33
Sicherheitsfrage	28
SimpleResponse	9, 17
Slots	6, <i>Siehe Slot</i>
Sperrung einer RFID-Karte	19
statusCode	10
Storno einer Buchung	32
System RFID-Karte	6
TAN	15, 21
TAN-Code	30
Terminal	26
terminalID	9
Türöffnung	24
Türstatus	24, 25
ValidationResponse	10
Zahlungsmittel	28