

## Texteingabefelder mit Hilfe von `UIPreferencesTextTableCell`

Geschrieben von: Philipp

WEDNESDAY, 27 FEBRUARY 2008 12:07 - Zuletzt aktualisiert TUESDAY, 04 MARCH 2008 02:41

---

Wer für seine Applikation eine Einstellungs-View mit Hilfe von `UIPreferencesTable` realisieren möchte, wird nicht um die Verwendung eines Texteingabefeldes drumherum kommen. Im folgenden wird erklärt, wie solch Texteingabefelder mit Hilfe von `UIPreferencesTextTableCell` realisiert werden:

Als erstes ersteinmal die benötigten Header importieren:

```
#import <UIKit/UIView.h>
#import <UIKit/UIPreferencesTable.h>
#import <UIKit/UIPreferencesTextTableCell.h>
```

Die Tabelle selber generieren (am besten in der *init* Methode der Klasse):

```
// das benötigte Rect
```

```
struct CGRect rect = [UIHardware fullScreenApplicationContentRect];
```

```
rect.origin.x = rect.origin.y = 0.0f;
```

```
// die benötigte view
```

```
UIView* mainView = [[UIView alloc] initWithFrame: rect];
```

```
// die benötigte UIPreferencesTable
```

```
UIPreferencesTable * table = [[UIPreferencesTable alloc] initWithFrame: CGRectMake(0.0f, 42.0f, rect.size.width, rect.size.height - 42.0f)];
```

```
[table setDataSource:self];
```

## Texteingabefelder mit Hilfe von UIPreferencesTextTableCell

Geschrieben von: Philipp

WEDNESDAY, 27 FEBRUARY 2008 12:07 - Zuletzt aktualisiert TUESDAY, 04 MARCH 2008 02:41

---

```
[table setDelegate:self];
```

```
// die table wird der view hinzugefügt
```

```
[mainView addSubview: titlebar];
```

Nun die neuen Zellen initialisieren - es werden hier drei Zellen erstellt, eine für normalen Text, eine als Passworтеingabefeld und eine welche nur Zahlen akzeptiert: // normal

```
UIPreferencesTextTableCell * textCell = [[UIPreferencesTextTableCell alloc]
initWithFrame:CGRectMake(                                0.0f, 0.0f, rect.size.width,
48.0f
)];
[
textCell
setTitle:
@"Text:"
];
[textCell setPlaceholderValue:@"erforderlich"];
[[textCell textField] setReturnType:4]; // "Weiter"

// password UIPreferencesTextTableCell * passwordCell = [[UIPreferencesTextTableCell
alloc] initWithFrame:CGRectMake(                                0.0
f
,
0.0f
,
rect
.size.width,
48.0f
)];
[
passwordCell
setTitle:
@"Passwort:"
];
[passwordCell setPlaceholderValue:@"erforderlich"];
```

## Texteingabefelder mit Hilfe von `UIPreferencesTextTableCell`

Geschrieben von: Philipp

WEDNESDAY, 27 FEBRUARY 2008 12:07 - Zuletzt aktualisiert TUESDAY, 04 MARCH 2008 02:41

---

```
[[passwordCell textField] setSecure:YES]; // Eingaben werden durch einen Punkt verschleiert

[[_passwordCell textField] setReturnKeyType:4]; // "Weiter"
```

```
// akzeptiert nur zahlen
UIPreferencesTextTableCell * numberCell = [[UIPreferencesTextTableCell alloc]
initWithFrame:CGRectMake(0.0f, 0.0f, rect.size.wid
th, 48.0f
)];
[
numberCell
setTitle:
@"Nummer"
];
[numberCell setPlaceholderValue:@"nur Zahlen"];
[[numberCell textField] setReturnKeyType:0]; // "Return"
```

```
[[numberCell textField] setPreferredKeyboardType:7]; // nur die Nummern-Tastatur in
schwarz
```

```
// Tabelle neu zeichnen
```

```
[table reloadData];
```

Zur Erklärung:

- mit *setTitle* wird der sichtbare Titel festgelegt
- mit *setValue* (im Code nicht aufgeführt) kann ein vorbelegter Text definiert werden
- *setPlaceholderValue* zeigt einen grauen Text, welcher sichtbar ist, wenn vom User noch kein Text eingegeben wurde, oder aber wenn mit *setValue* kein Text vorbelegt wurde

- mit *textField* kann sich ein Object vom Type *UITextField* geholt werden, dadurch können u.a. Eigenschaften für das Keyboard festgelegt werden

- bei *setSecure:YES* wird das *UITextField* der Zelle als ein Passwortfeld festgelegt, da jetzt Texteingaben durch einen Punkt verschleiert werden

- *setPreferredKeyboardType* definiert den zu verwendenden Keyboard-Typ - folgende Angaben sind dabei möglich:

- 0 = Standard
- 1 = Zahlen mit Punktnotationen
- 2 = Zahlen wie beim Telefon KeyPad
- 3 = zur Eingabe von URL's
- 4 = zur Eingabe vom SMS
- 5 = das gleiche wie bei 0, jedoch in schwarz
- 6 = das gleiche wie bei 2, jedoch ohne die eingabe von "+" oder "\*" etc.
- 7 = wie 6 jedoch in schwarz
- 8 = zur Eingabe von Email-Adressen

- 9 = zur Eingabe von Email-Adressen, jedoch ist die Leerraum-Taste mit dem "@" belegt
  
- `setReturnKeyType` definiert das Aussehen der Return-Taste auf dem Keyboard (falls vorhanden) - folgende Angaben sind dabei möglich (Die Textangaben variieren je nachdem welche Sprache auf dem iPhone aktiviert ist):
  - 0 = "Return" in grau
  - 1 = "Go" in blau
  - 2 = "Google" in blau
  - 3 = "Join" in blau
  - 4 = "Next" in grau
  - 5 = "Route" in blau
  - 6 = "Search" in blau
  - 7 = "Send" in blau
  - 8 = "Yahoo" in blau