

Package game

```
import "tic-tac-toe/game"
```

[Overview](#)[Index](#)

Overview ▾

Das Paket 'game' beinhaltet die Tic Tac Toe Spiellogik.

Index ▾

```
type Move
func (mv *Move) allowed(ttt TicTacToe) bool
type TicTacToe
func (ttt *TicTacToe) Play(mv Move)
func (ttt *TicTacToe) Reset()
func (ttt *TicTacToe) end() bool
func (ttt *TicTacToe) player()
func (ttt *TicTacToe) set(mv Move)
func (ttt *TicTacToe) win(mv Move) bool
```

Package files

move.go tictactoe.go

type Move

Move repräsentiert einen Spielzug und ist zugleich das API-Anfrage-Format.

```
type Move struct {
    Player int  `json:"player"`
    Column int  `json:"col"`
    Row    int  `json:"row"`
    Reset  bool `json:"reset"`
}
```

func (*Move) allowed

```
func (mv *Move) allowed(ttt TicTacToe) bool
```

allowed prüft, ob der Zug zulässig ist bzw. ob das Feld bereits belegt ist und gibt einen Wahrheitswert (bool) zurück.

type TicTacToe

TicTacToe repräsentiert den Spielstand und ist zugleich das API-Antwort-Format.

```
type TicTacToe struct {  
    Finished    bool    `json:"finished"`  
    Winner      int     `json:"winner"`  
    NextPlayer  int     `json:"nextPlayer"`  
    Field       [3]string `json:"field"`  
}
```

func (*TicTacToe) Play

```
func (ttt *TicTacToe) Play(mv Move)
```

Play stellt den Spiel-Ablauf dar.

func (*TicTacToe) Reset

```
func (ttt *TicTacToe) Reset()
```

Reset setzt die übergebene Variable auf Spielbeginn.

func (*TicTacToe) end

```
func (ttt *TicTacToe) end() bool
```

end überprüft, ob das Spielfeld vollständig belegt ist.

func (*TicTacToe) player

```
func (ttt *TicTacToe) player()
```

player bestimmt den nächsten Spieler und setzt diesen Wert in der übergebenen Variable.

func (*TicTacToe) set

```
func (ttt *TicTacToe) set(mv Move)
```

set überträgt den Spielzug auf das Spielfeld.

func (*TicTacToe) win

```
func (ttt *TicTacToe) win(mv Move) bool
```

win überprüft, ob ein Spieler gewonnen hat und gibt dies als bool zurück.
