From bound in *futures* crate

```
impl<T> Sink for Sender<T>
```

```
type SinkItem = T
```

The type of value that the sink accepts.

```
type SinkError = SendError<T>
```

The type of value produced by the sink when an error occurs.

```
[-]fn start_send(&mut self, msg: T) -> StartSend<T, SendError<T>>
```

Begin the process of sending a value to the sink. Read more

Trait futures::stream::Stream

```
[-]fn forward<S>(self, sink: S) -> Forward<Self, S>
    where S: Sink<SinkItem=Self::Item>,
          Self::Error: From<S::SinkError>,
          Self: Sized
```

Type inference

```
18 fn main() {
19     let mut route = Vec::new();
20     route.push(Meters(2000.0).into());
21     route.push(Miles(2.0));
22     route.push(From::from(Meters(3000.0)));
23
24     println!("{:?}", route);
25 }
```